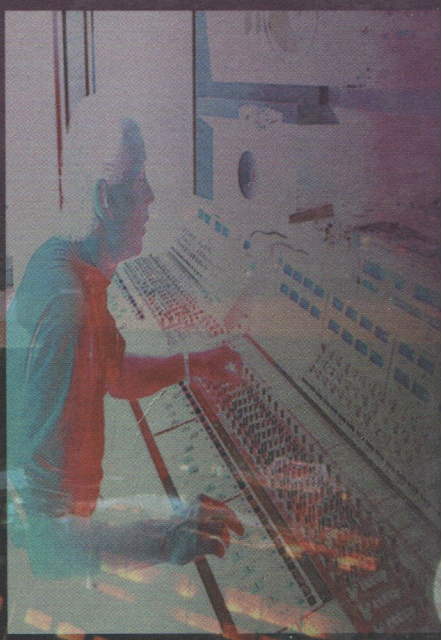
COMPRESIA SEMNALELOR

Codarea vorbirii, a semnalelor audio, a textului, a imaginii și a semnalelor video



Editor
Nikil Jayant
Bell Laboratories, USA

Teora

Internet: www.teora.ro

COMPRESIA SEMNALELOR

**Codarea vorbirii, a semnalelor audio,
a textului, a imaginii și a semnalelor video**

COMPRESIA SEMNALELOR

Codarea vorbirii, a semnalelor audio,
a textului, a imaginii si a semnalelor video

Cuprins

COMPRESIA SEMNALELOR 7
Nikil Jayant

METODE CURENTE DE CODARE A VORBIRII 18
Richard V. Cox

CODAREA AUDIO ISO/MPEG 69
Peter Noll şi Davis Pan

COMPRESIA SEMNALELOR VIDEO
ŞI DE IMAGINE: TRECERE ÎN REVISTĂ 117
Christine I. Podilchuk şi Robert J. Safranek

COMPRESIA FĂRĂ PIERDERI PENTRU TEXT ŞI IMAGINI 172
Alistair Moffat, Timothy C. Bell şi Ian H. Witten

Nota traducătorului

Dorim să menționăm de la început câteva distincții făcute în carte, care sunt fundamentale pentru înțelegerea conținutului acesteia.

1. Semnalele sonore de *vorbire* și *audio*. Primele reprezintă numai tipul de comunicare sonoră prin vorbire (cazul specific al comunicării telefonice), în timp ce al doilea reprezintă semnalul sonor în sensul cel mai larg (de exemplu muzică).
2. Semnalele de *imagine* și *video*. Prima noțiune semnifică de fapt imaginea statică, poză, în timp ce „video" implică dinamica, filmul, mișcarea.
3. *Codarea semnalului* și *codarea canalului*. Prima se referă la codarea informației utile (semnalul de vorbire, audio, video sau imagine) și a mai fost numită și codare sursă. A doua se referă la conversia semnalului codat într-o manieră care să-l facă potrivit pentru trimiterea lui prin canalul de transmisie. Aceasta implică în general introducerea unor protecții la erori de canal, modulare etc.

Notele de subsol ale autorilor s-au notat, ca și în textul original, cu [a], [b], [c] etc.

Notele de subsol ale traducătorului s-au notat cu „[NT1]", „[NT2]", „[NT3]" etc.

Lucrările din bibliografie, la care se face referire în text, au fost notate fie cu numărul din listă atașat textului în același mod ca notele de subsol (de exemplu [2, 3, 54]), fie cu numărul din listă între paranteze drepte (de exemplu [34], [45]).

Am încercat să păstrăm abrevierile din limba engleză deoarece acestea se repetă în multe alte lucrări (cele mai multe netraduse încă în limba română) și au un anumit grad de consacrare.

Unii termeni au fost păstrați și în limba engleză, în general între paranteze și cu litere italice. Am considerat acest lucru util pentru cititorii care vor aborda, sau au citit deja, alte lucrări cu subiect similar în limba engleză. Multe dintre cuvintele din această categorie sunt noi chiar în limba engleză, create parcă numai pentru aceste aplicații recente și nu pot fi găsite în dicționarele uzuale, ba nici chiar în unele dicționare tehnice engleze (*wavelet*, *runlength*, *cepstrum*, *aliasing* etc.). O altă categorie de cuvinte este aceea în care sensul original al cuvântului are o legătură numai sugestivă și indirectă cu contextul în care este utilizat și de aceea echivalența între termenul original și traducerea lui nu este totdeauna evidentă, legându-se mult de contextul particular. Uneori termenii sunt deja uzuali în contextul tehnic (de exemplu *offset*), alteori însă nu (*toll quality*).

Titlurile cărților prezentate în Referințe bibliografice la sfârșitul fiecărui capitol au fost păstrate în limba engleză, deoarece lucrările nu au fost publicate în limba română și echivalentele românești ale titlurilor ar fi mai puțin utile decât originalele.

Articol din „International Journal of High Speed Electronics and Systems", Vol. 8, No. 1 (1997) 1-12
© World Scientific Publishing Company

COMPRESIA SEMNALELOR

NIKIL JAYANT
Lucent Technologies, Bell Laboratories
Murray Hill, New Jersey 07974, USA

Acest articol este o introducere la o lucrare specială despre codarea și compresia semnalelor. Începem prin a defini conceptele de codare digitală și compresie a semnalelor audio-vizuale. Apoi vom descrie cele patru dimensiuni ale performanței codării: rata de bit, calitatea semnalului, întârzierea de procesare și complexitatea. Vom ilustra cele două principii de bază ale codării audio-vizuale, eliminarea redundanței semnalului și adaptarea sistemului de cuantizare la proprietățile sistemului de percepție uman, cu exemple specifice recente de algoritmi de codare. Vom enumera standardele și aplicațiile referitoare la compresia semnalelor audio-vizuale. O aplicație cu o evoluție rapidă este vehicularea prin Internet a informațiilor audio-vizuale, un domeniu prea nou pentru a putea fi acoperit în articolele din această culegere. Vom încheia articolul cu prezentarea opiniilor noastre asupra direcțiilor viitoare de cercetare în acest domeniu.

1. Procesarea și codarea digitală a semnalelor

Articolele din această culegere se referă la codarea digitală pentru reducerea ratei de bit. Această disciplină s-a dezvoltat în cel puțin două școli din domeniul comunicațiilor digitale.

Compresia semnalelor de date este o parte integrantă a teoriei informației. În particular, domeniul teoriei ratei de distorsiuni se ocupă de diferite probleme calitative ale codării fără pierderi și cu pierderi, folosind conceptul de entropie a sursei în proces.

Compresia semnalelor a constituit, de asemenea, un subiect principal al procesării vorbirii și imaginilor. Codarea, sinteza și recunoașterea sunt trei domenii cheie de interes în procesarea semnalelor audio-vizuale. În cadrul *codării*, scopul este de a obține o reprezentare digitală compactă a semnalului pentru economia transmisiei sau a stocării. O atenție deosebită în *sinteză* se dă creării semnalelor corespunzătoare vorbirii sau informației vizuale, pornind mai degrabă de la *text* decât de la vorbirea umană sau de la o imagine reală, ca în cazul codării. Utilizatorul informațiilor audio-vizuale codate sau sintetizate este omul. Pentru recunoaștere, atenția se îndreaptă asupra înțelegerii de către mașină (calculator) a conținutului informației din cadrul semnalului, de obicei pentru a ajuta la îndeplinirea unor sarcini. Fiecare dintre aceste domenii ale procesării semnalelor audio-vizuale este în sinergie cu alte domenii. Astfel, de exemplu, codarea și sinteza sunt folosite pentru a construi înregistratoare digitale pe bandă pentru sisteme care răspund la voce; sinteza și recunoașterea asigură sisteme de intercomunicare pentru mașini controlate prin voce; iar recunoașterea și codarea dau informații suplimentare într-o codare cu o rată de bit foarte redusă, ca în situația localizării feței într-o imagine „cap și umeri" într-o scenă de videofon, urmată de o codare mult mai atentă a trăsăturilor feței.

Atenția este îndreptată, în acest caz special, în întregime asupra domeniului codării. Articolele care urmează acestei introduceri[1-3] se ocupă în special de codarea vorbirii, a semnalelor audio de bandă largă și a semnalelor de imagine. Articolele dedicate codării imaginii analizează atât cazul imaginilor statice cât și pe al celor în mișcare (video).

Ultimul capitol[4] se ocupă de compresia de date. Acest subiect este important prin sine însuși. Metodele de compresie de date fără pierderi sau cu pierderi sunt o componentă semnificativă a sistemelor pentru compresia semnalelor. Exemple sunt codoarele continue pentru semnale de fax și codoarele fără pierderi, care urmează sistemelor de cuantizare pentru semnale audio-vizuale.

Ultima parte a acestui articol se concentrează asupra semnalelor audio-vizuale, în încercarea de a da o imagine de ansamblu a modurilor de tratare din cele trei articole care urmează.

2. Compresia semnalelor audio-vizuale: codarea vorbirii, a semnalului audio și a imaginii

Subiectul compresiei de semnal a fost descris foarte amănunțit în câteva texte[5-7] și prin articole de trecere în revistă[8,9] relativ recente, cititorul fiind sfătuit să le consulte pentru discuții detaliate despre principiile codării, precum și despre activitățile de standardizare în acest domeniu. În acest capitol vom realiza o scurtă, dar cuprinzătoare și adusă la zi, trecere în revistă a stadiului în care se află codarea semnalelor, pentru a putea înțelege noutățile detaliate care vor fi expuse în cele trei lucrări care urmează acestui articol introductiv.

Cei patru parametri fundamentali ai codării semnalelor sunt *calitatea semnalului, rata de bit, întârzierea de procesare* și *complexitatea* implementării. Primul deziderat al codării este de a descrește rata de bit menținând în același timp un nivel specificat de calitate a semnalului. În general, este necesar să se mențină și un nivel specificat al întârzierii procesării și al complexității implementării.

Deși discursul din acest articol este inspirat de semnalele audio-vizuale, dimensiunile performanțelor despre care vom discuta se aplică în aceeași măsură semnalelor de date. Ceea ce este general acceptat este faptul că transmisiile de date se presupun a fi fără pierderi, implicând din punct de vedere matematic o calitate perfectă a semnalului. În plus, dimensiunea întârzierii de procesare este în general mult mai puțin importantă pentru date decât pentru semnalele audio-vizuale.

2.1. *Rata de bit*

Tabelul 1 definește lărgimile de bandă și ratele de eșantionare tipice în comunicațiile audiovizuale. Rata de eșantionare este de cel puțin două ori mai mare decât lărgimea de bandă, în conformitate cu teorema lui Nyquist. În cazul semnalelor video, lărgimea de bandă este în general o mărime implicită.

Rata de bit se exprimă în general în *biți pe secundă* (bps) sau în *biți pe eșantion*. Numărul de biți pe secundă este pur și simplu produsul dintre rata de eșantionare (măsurată în Hertzi sau pixeli pe secundă) și numărul mediu de biți pe eșantion, utilizat în sistemul de cuantizare al codorului.

De exemplu, pe baza ratelor de eșantionare prezentate în tabelul 1, 8 biți pe eșantion în codarea semnalului de vorbire telefonică vor implica o rată de bit generală de

64 kbps, iar 16 biți pe eșantion în codarea fiecăruia dintre cele două canale audio (stânga și dreapta) în formatul CD-stereo vor implica o rată generală de 1410 kbps. O rată *medie* de 0,25 biți pe eșantion în cuantizarea coeficienților de transformare într-un codor HDTV va conduce la o rată de 15 Mbps. Cele mai multe exemple de rată de bit coborâtă la codare implică alocări de bit inegale pentru componentele semnalului rezultate din analiza timp-frecvență, iar ratele medii în gama 0,2-2,0 biți pe eșantion sunt relativ frecvente în practică.

Tabelul 1. Niveluri standard ale semnalelor audio și video

Format audio	Rata de eșantionare	Gama de frecvență
Telefonie	8 kHz	200-3400 Hz
Teleconferințe	16 kHz	50-7000 Hz
Compact disc (CD)	44,1 kHz	20-20000 Hz
Bandă magnetică digitală	48 kHz	20-20000 Hz
Formate video	**Rata de eșantionare**	**Rezoluția spațio-temporală**
CIF	3 MHz	360×288×30
CCIR	12 MHz	720×576×30
HDTV	60 MHz	1280×720×60

CIF = Format intermediar comun
CCIR = Comitetul Consultativ Internațional de Radio
HDTV = Televiziune de înaltă definiție

Câteva sisteme de codare folosesc, de asemenea, codarea cu rate variabile (în mod ideal, transmisie cu calitate constantă) pentru părți diferite ale semnalelor audio-vizuale nestaționare. Codarea cu rate variabile este în mod particular potrivită pentru transmisiile pe pachete.

2.2. *Calitatea semnalului*

Dat fiind că judecătorul final al sistemului de compresie a semnalului este observatorul uman, calitatea semnalului audio-vizual este descrisă cel mai bine pe baza unor criterii subiective. Este larg acceptată o scară cu cinci trepte a *calității* semnalelor, la care se adaugă uneori măsurători de *inteligibilitate*. Măsurarea subiectivă a calității vorbirii sau a imaginii într-o manieră repetabilă și consecventă este o chestiune dificilă și care dă multă bătaie de cap. Măsurătorile calității compozite ale unui semnal audio-vizual reprezintă o problemă cel puțin la fel de dificilă, iar datele despre acest subiect sunt sărace.

În cazul semnalelor de imagine, calitatea apreciată subiectiv trebuie să fie exprimată ca o funcție de distanța de la care se privește. Este acceptată ca tipică o distanță echivalentă cu de patru ori înălțimea imaginii.

2.3. *Întârzierea procesării*

Întârzierea procesării este suma întârzierilor care apar în procesele de codare și decodare. La codare, întârzierea este introdusă în procesul de stocare tampon a datelor pentru a fi analizate în vederea reducerii redundanței. Ca exemple pot fi date metodele de procesare-bloc a transformărilor liniare și cuantizarea vectorilor. La decodare, întârzierile sunt introduse de operații bazate pe blocuri, cum ar fi transformările inverse și de operații ca interpolarea, pentru a mări frecvența cadrelor în cazul imaginilor.

O altă sursă de întârziere a transmisiei în comunicații se datorează digitalizării semnalelor folosite în rețea. Aplicații cum sunt telefonia și videoconferințele necesită o valoare minimă a întârzierilor, în funcție de necesitățile algoritmului de compresie, pentru a asigura un nivel adecvat al calității semnalului. În comunicațiile unidirecționale, cum ar fi radiodifuziunea, întârzierea este mai puțin importantă, deși este încă necesară o minimizare a întârzierilor de procesare, atunci când ne referim la cerințele impuse de stațiile de comutare. În aplicațiile de stocare a informațiilor, întârzierea codării este total irelevantă, atâta timp cât întârzierea decodării este suficient de mică pentru a asigura o bună calitate a serviciilor.

2.4. *Complexitatea*

Complexitatea este măsurată atât prin procesele aritmetice necesare algoritmului (de obicei măsurate în „mips" – milioane de instrucțiuni pe secundă), cât și prin cerințele de memorie (măsurate de obicei în kiloocteți de ROM sau RAM). Folosirea mips ca măsură a complexității este în mod particular adecvată pentru implementări de uz general, sau pentru calculatoare. În aplicații specifice de tipul circuitelor integrate (ASIC = *Application Specific Integrated Circuits*) și alte măsuri ale complexității (ca, de exemplu, numărul de tranzistoare sau numărul de porți) pot fi folositoare. Complexitatea este un parametru important de performanță din cel puțin două motive – nevoia de a minimiza costul și cerința de minimizare a puterii disipate (în mod special în aplicațiile cu dispozitive portabile).

În aplicații de tipul transmisiunilor radio sau TV, este în mod special importantă minimizarea complexității decodorului. Aceasta, în timp ce complexitatea codorului este cu mult mai puțin importantă, sau chiar nesemnificativă.

Disponibilitatea de complexitate (de integrare) per-cip (atât în ceea ce privește prelucrarea aritmetică cât și în memorie) a crescut exponențial de-a lungul ultimilor câțiva ani, fără a se întrevedea o tendință de saturare. Aceasta va permite utilizarea în practică a algoritmilor cu grad de sofisticare din ce în ce mai crescut la codificarea semnalelor (ca și pentru alte tehnologii multimedia).

2.5. *Algoritmi de codare*

Este necesară o varietate de sisteme de codare pentru a se asigura compromisul optim între întârziere și complexitate. Peste toate acestea însă, există numai două principii fundamentale de compresie: (a) eliminarea redundanței de ordin statistic (sau determinist) din sursa de semnal și (b) adaptarea sistemului de cuantizare la proprietățile sistemului de percepție uman. În cazul semnalelor de date, compresia se bazează în întregime pe eliminarea redundanței.

Semnalele vorbirii au un model universal foarte bine cunoscut de a se produce, ceea ce face posibilă o tehnică eficientă de reducere a redundanței: în primul rând, codarea cu predicție liniară în domeniul timp. În principiu, structura semnalului poate fi, de asemenea, exploatată prin paradigma alocării inegale de bit în domeniul frecvență și aceasta se potrivește în particular în codarea audio în sens mai larg.

Semnalele de imagine și video (dinamice) sunt în mod evident redundante pe linii și pe cadre. Această remarcă este foarte evidentă în cazul faxului de înaltă rezoluție, unde structura apare în forma unor mănunchiuri semnificative de pixeli negri pe fond alb, în două dimensiuni (2-D), ceea ce se exploatează prin diferite forme de codare cu

lungime variabilă. Structura semnalului este, de asemenea, foarte evidentă în cazul scenelor de videoconferințe, în care predomină busturi cu evoluții spațio-temporale reduse.

Foarte utilizată este codarea MC-DCT pentru semnale video, care utilizează un hibrid de predicție (codarea cu predicție de la un cadru la altul sub forma *compensării mișcării)* și o operare în domeniul frecvență, pentru cuantificarea predicției erorilor (*transformarea cosinus discretă 2-D).*

În codarea vorbirii, ca și în codarea imaginii și a semnalelor video, sistemul de cuantizare poate fi avantajat în realizarea unei reduceri chiar mai mari a ratei de bit – pentru un nivel dat de calitate a semnalului – de un fenomen fiziologic, numit mascarea zgomotului. Mascarea este fenomenul prin care un stimul puternic (semnalul util) acoperă complet un semnal mai slab (zgomotul de cuantizare) în vecinătatea lui spectrală sau spațio-temporală. Cu alte cuvinte, nivele matematic semnificative de distorsiuni de cuantizare pot fi acceptate cu o diminuare minimă sau chiar nulă a calității percepției semnalului. Forma cea mai comună de codare adaptată la percepție este cea în care componente selectate cu grijă dintr-o transformare în domeniul frecvență (spectrele semnalelor cu durată mică) sunt fie cuantizate grosier, fie complet ignorate. Progresul cel mai mare în codarea perceptuală a fost realizat în domeniul audio extins. Spre deosebire de voce sau de semnalele sistemelor videofon (videotelefonie), domeniul audio extins nu are un model universal de redundanță, sau repetare a cadrelor pentru reducerea redundanței. Prin compresia acestor semnale se urmărește reducerea redundanței cât mai mult posibil, prin tehnici clasice de predicție și codare adaptativă, cât și transferarea celei mai mari părți a strădaniilor în zona perceptuală a lucrurilor (unde se realizează mascarea zgomotului). Câștigul codării perceptuale este maxim atunci când rata de bit este suficientă ca să producă o codare critică, sau la limită. În codarea audio, ilustrată în figura 1, această limită este descrisă de semnalul specific JND (*Just Noticeable Distortion* = distorsiuni abia perceptibile), ca o funcție de frecvență. Singurele componente salvate în compresia unui bloc audio dat sunt cele care se extind în afara semnalului JND de mai sus pentru spectrul de putere de scurtă durată. Codorul audio perceptual (PAC = *Perceptual Audio Coder*) realizează niveluri foarte ridicate de compresie, care se apropie de 20:1. O realizare deosebită este compresia celor 1410 kbps ai semnalului CD-stereo la o rată de bit de ordinul a 64 kbps cu prețul unui sacrificiu nesemnificativ al calității semnalului[9].

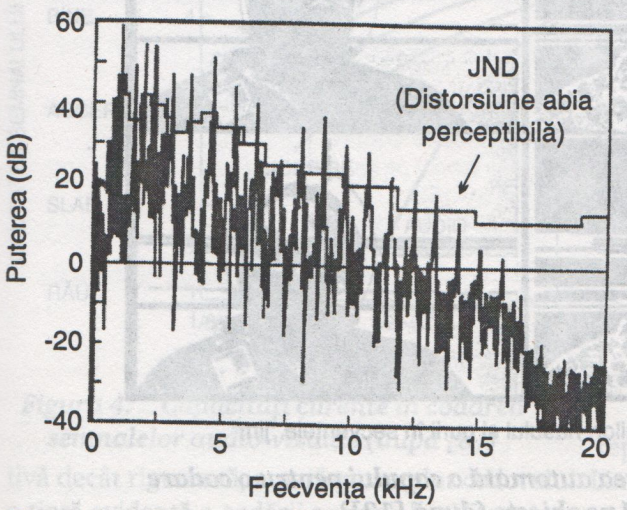
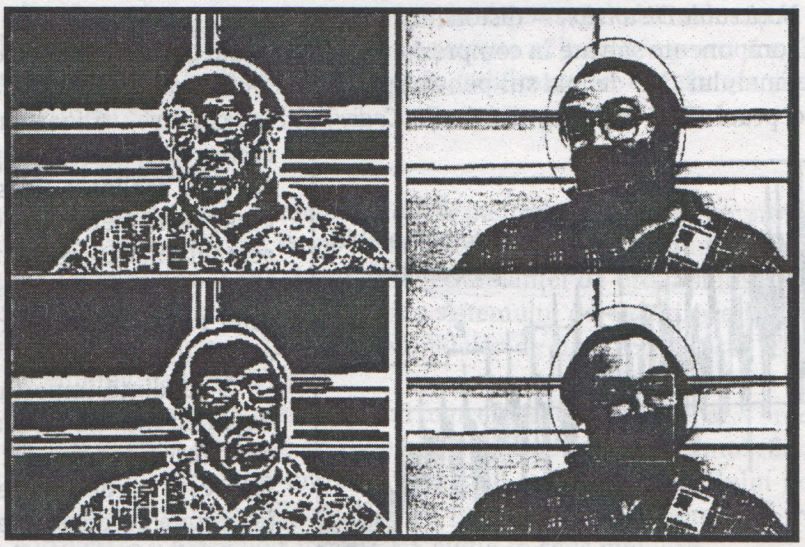


Figura 1. Codarea audio perceptuală (după [9])

O tendință importantă în codarea cu rată de bit foarte joasă, este folosirea modelării, care folosește structuri de semnale împreună cu considerații asupra percepției. De exemplu, pentru o scenă cu un bust sau prim-plan (*head-and-shoulder*), obiectul-față poate fi procesat special

pentru a fi mai bine reprodus, cu prețul unei scăderi ușoare a performanțelor pentru redarea fundalului, mai puțin semnificativ din punct de vedere perceptual. Aceasta se poate obține fie prin procesarea implicită a feței, făcând ca un vector de cuantizare cu adaptare la scenă să schimbe modul de codare pentru obiectul dominant sau din prim-plan[11], fie printr-un algoritm explicit care modelează fața și capul prin forme convenționale ca o elipsă și un dreptunghi (figura 2) și extinde rata de bit în interiorul elipsei și/sau al dreptunghiului.[12]

Locații ale ochilor, nasului și gurii în secvențele "elena" și "roberto"



Locații ale ochilor, nasului și gurii în secvențele "jim"

Figura 2. Identificarea automată a capului pentru o codare bazată pe obiecte (după [12])

2.6. *Nivelul de vârf atins în domeniu (state of the art)*

Figurile 3 şi 4 descriu capacităţile actuale de compresie a semnalului.

Figura 3 arată gama de aplicaţii pe care le poate susţine capacitatea actuală de compresie a semnalelor. Aplicaţii cu o rată de bit sub 10 kbps, cum sunt vorbirea protejată (cifrată), radio celular, transmisia prin poştă electronică a vocii şi imaginii, sunt relativ uşor de realizat practic. Între 10 şi 20 kbps aplicaţiile ajung în aria telefoniei şi teleconferinţei. Între 20 şi 100 kbps se plasează câteva aplicaţii audio-vizuale incluzând grafica, video Internet, voce prin Internet, muzică la nivel de radiodifuziune. Videoconferinţele devin interesante la o rată de bit între 100 şi 500 kbps, filmele la cerere (televiziune interactivă) necesită 1 Mbps, iar televiziunea de înaltă definiţie (HDTV) 20 Mbps.

Figura 3 evidenţiază ratele de bit (în kbps şi Mbps) care corespund claselor generice de aplicaţii curente. Comunicarea în bune condiţii (cotată cu minim 3,5) este presupusă de multe dintre aceste aplicaţii. Vorbirea de calitate la nivelul reţelelor implică o cotare de cel puţin 4, în timp ce calitatea necesară radiodifuziunii audio (şi video) necesită în mod ideal o cotare în zona lui 4,5.

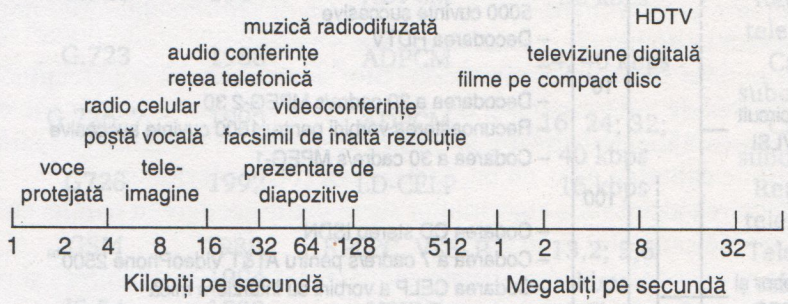


Figura 3. Aplicaţii ale compresiei semnalelor (după [8])

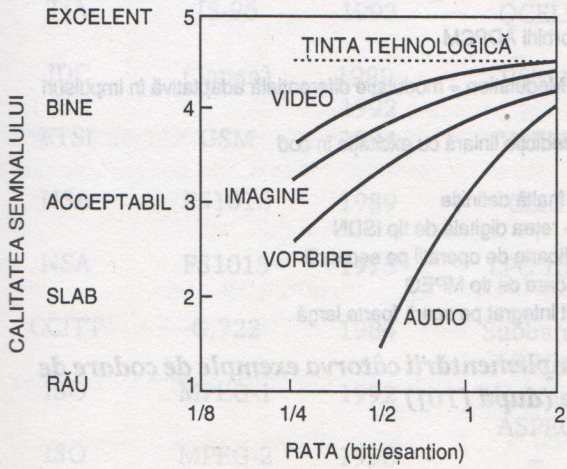


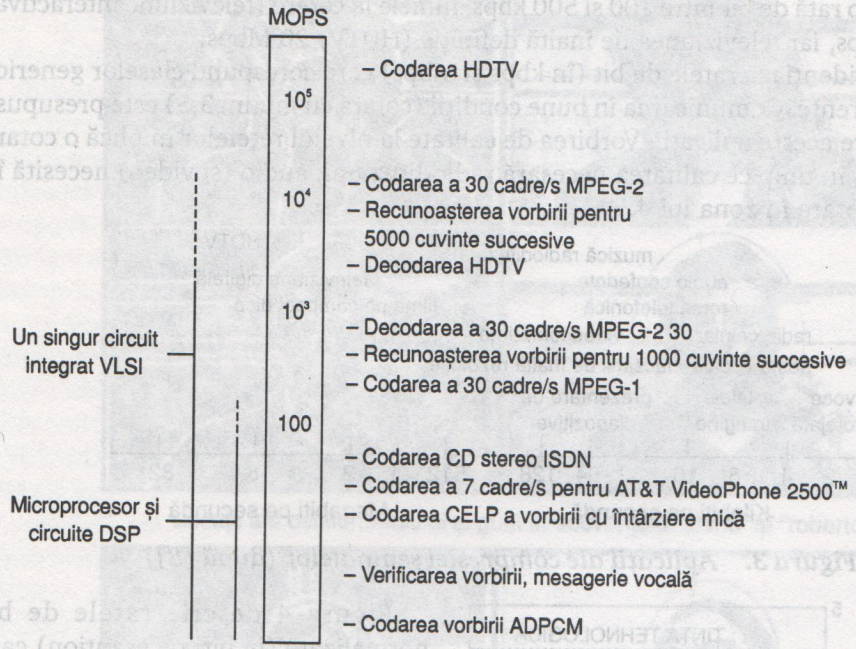
Figura 4. Capacităţi curente în codarea semnalelor audio-vizuale (după [8])

Figura 4 descrie ratele de bit normalizate (în biţi pe eşantion) care pot suporta diferite niveluri de calitate a semnalului, pe o scară cu 5 niveluri. Aceste numere pot fi convertite în rate absolute (în kbps sau Mbps) dacă se înmulţeşte rata de bit normalizată cu frecvenţa de eşantionare din tabelul 1. Rezultatele din figura 4 sunt de asemenea conservative, cele patru curbe din figură se vor îndrepta către limita ideală a tehnologiei, pe măsură ce tehnica de compresie devine mai sofisticată.

Figurile 3 şi 4 trebuie privite ca o reprezentare mai degrabă aproximativă decât riguroasă, a performanţelor codorului. Linia orizontală din figura 4 reprezintă o ţintă evidentă a codării semnalelor. Viteza cu care progresăm în atingerea acestei ţinte este în general dependentă de semnalul de intrare. Mai trebuie notat şi faptul că există

limite fundamentale în codarea semnalelor, determinate de *entropia perceptuală*, cum ar fi rata minimă de bit pentru un anumit nivel de calitate a semnalului. Entropia perceptuală este o funcție care depinde de semnalul de intrare și de mecanismul de percepție umană.[10]

Figura 5 extinde perspectiva asupra tehnologiei, prin includerea dimensiunii complexității. În figură sunt incluse câteva exemple ilustrative de compresie, utilizate în domeniul multimedia. Tot aici sunt prezentate, cu aproximație, domeniile procesării de semnal de uz general sau specifice unor aplicații, ca și câteva aplicații din domeniul recunoașterii vorbirii.



ADPCM = Adaptive Differential Pulse Code Modulation = modulație diferențială adaptativă în impulsuri
CD = Compact disc
CELP = Code-Excited Linear Prediction = predicție liniară cu excitație în cod
DSP = Digital Signal Processor
HDTV = High-Definition TV = televiziune de înaltă definiție
ISDN = Integrated Digital Service Network = rețea digitală de tip ISDN
MOPS = Million operations per second = milioane de operații pe secundă
MPEG = Motion Picture Experts Group = codare de tip MPEG
VLSI = Very Large Scale Integration = circuit integrat pe scară foarte largă

Figura 5. Complexitatea necesară implementării câtorva exemple de codare de semnale (după [10])

2.7. *Standarde de codare*

Tabelul 2 recapitulează o serie de standarde pentru codarea semnalelor audio-vizuale. Prima grupă de standarde este pentru banda audio din telefonie, cu aplicații în rețeaua telefonică, în comunicațiile celulare și în cifrarea vocii. A doua grupă include standarde pentru telefonia de bandă largă și un standard audio. Ultima grupă include standarde atât pentru imagini statice cât și pentru domeniul video, la câteva rate de bit.

Algoritmii standardizaţi, prin definiţie rămân în urmă faţă de disponibilităţile curente de cercetare, dar şi datorită naturii procesului de standardizare. Noutăţile şi îmbunătăţirile brevetate sunt adesea utilizate pentru a ameliora un algoritm standardizat, prin încorporarea elementelor noi sub forma unor preprocesări ale codării şi corelarea cu decodorul standard şi cu sintaxa fluxului de biţi. Ca exemple, putem menţiona folosirea în standardul MPEG a unui codor psihoacustic asemănător cu PAC (codor audio perceptual) şi detectarea feţei ca preprocesare în standardul video CCITT sau MPEG.

Tabelul 2. Standarde pentru codarea vorbirii, a imaginilor şi a semnalelor video (după [1] şi [10])

Grupa de standarde	Standard	Anul	Algoritmul	Rata de bit	Aplicaţie
CCITT	G.711	1972	Legile μ şi A-PCM	64 kbps	Reţeaua telefonică
CCITT	G.721	1984	ADPCM	32 kbps	Reţeaua telefonică
CCITT	G.723	1988	ADPCM	24, 40 kbps	Cablu suboceanic
CCITT	G.726, 7	1990	ADPCM	16; 24; 32; 40 kbps	Cablu suboceanic
ITU-T	G728	1992	LD-CELP	16 kbps	Reţeaua telefonică
ETSI	GSM	1988, 1994	RPLPC, VSELP	13,2; 5,6 kbps	Telefonie celulară
TIA	IS-54	1989	VSELP	8 kbps	Telefonie celulară
TIA	IS-96	1993	QCELP	0,8; 2; 4; 8,5 kbps	Telefonie celulară
JDC	(Japan)	1989, 1992	VSELP	8; 4 kbps	Telefonie celulară
ETSI	GSM	1994	VSELP	5,6 kbps	Telefonie celulară
NSA	FS1016	1989	CELP	4,8 kbps	Secretizarea vocii
NSA	FS1015	1975	LPC 10E	2,4 kbps	Secretizarea vocii
CCITT	G.722	1984	Subband-ADPCM	32-64 kbps	Teleconferinţe
ISO	MPEG-1	1992	Musicam/ ASPEC	128-384 kbps	Înregistrare audio stereo
ISO	MPEG-2	1996	–	320; 384 kbps	Înregistrare audio – 5 căi
ISO	JBIG	1991	Cod cu lungime variabilă	0,05-0,10 bpp	Imagini digitale

Tabelul 2. (continuare)

Grupa de standarde	Standard	Anul	Algoritmul	Rata de bit	Aplicație
ISO	JPEG	1991	DCT	0,25-8,0 bpp	Imagini statice
ISO	MPEG1, 2	1991, 1994	MC-DCT	1-8 Mbps	Video adresabil
CCITT	Px64	1991	MC-DCT	64-1536 kbps	Videoconferințe
FCC	HDTV	1995	MC-DCT	20 Mbps	TV avansată

3. Tendințe în compresia semnalelor

Așa cum se va vedea și în articolele care urmează, domeniul compresiei de semnale este deosebit de dinamic și la un nivel avansat de maturitate. Este însă încă loc de evoluții viitoare și vom menționa mai jos câteva dintre direcțiile mai importante, cu referire directă la domeniul semnalelor audio-vizuale:

- continuarea înțelegerii caracteristicilor semnalelor și a percepției umane, în încercarea de a determina și de a atinge limitele fundamentale ale codării,
- cercetarea aplicată pentru a identifica elemente interesante în spațiul 4-D, care cuprinde calitatea, rata de bit, întârzierea și complexitatea,
- proiectarea avansată a algoritmilor sursă pentru codare în paralel cu tehnicile de codare ale canalelor și, în sfârșit,
- proiectarea codării audio-vizuale corelate cu tehnicile de sinteză și recunoaștere pentru produsele și serviciile multimedia.

Referințe bibliografice

1. R. V. Cox, "Current methods of speech coding", volumul de față.
2. P. Noll și D. Pan, "ISO-MPEG audio coding", volumul de față.
3. C. Podilchuk și R. J. Safranek, "Image and video compression: A review", volumul de față.
4. A. Moffat, T. C. Bell și I. H. Witten, "Lossless compression for text and images", volumul de față.
5. N. S. Jayant și P. Noll, *Digital Coding of Waveforms*, Prentice Hall, 1984.
6. A. N. Netravali și B. Haskell, *Digital Images*, Plenum Press, 1987.
7. P. E. Papamichalis, *Practical Approaches to Speech Coding*, Prentice Hall, 1987.
8. N. S. Jayant, "Signal compression: Technology targets and research directions", *IEEE J. Selected Areas in Communications* (1992) 796–818.
9. N. S. Jayant, J. D. Johnston și R. S. Safranek, "Signal compression based on models of human perception", *Proc. IEEE*, Oct. 1993, pp. 1385–1422.
10. N. S. Jayant, V. B. Lawrence, B. Ackland și L. R. Rabiner, "Technological dimensions of multimedia", *AT&T J.*, Sept. 1995, pp. 14–33.
11. D. Wang și J. Hartung, "Codebook adaptation algorithm for a scene-adaptive video coder", *Proc. ICASSP '95*, Detroit, May 1995.
12. A. Eleftheriadis și A. Jacquin, "Automatic face location detection for model-assisted rate control in H.261-compatible coding of video", în *Image Communication* 7 (1995) 435–455.

Dr. Nikil Jayant este director al Laboratorului de Cercetare pentru Comunicații Multimedia de la Bell Labs., ramura R&D a Lucent Technologies (până de curând Systems and Technologies, parte a AT&T). În această calitate, dr. Jayant este responsabil cu crearea și comercializarea tehnologiilor pentru comunicații audio-vizuale și sisteme informatice multimedia.

Înainte, la AT&T Bell Laboratories, dr. Jayant a creat și condus Departamentul de Cercetare a Procesării de Semnale și departamentul de Tehnologii Audio Avansate. Cercetările personale ale dr. Jayant au fost în domeniul codării digitale și transmisiei semnalelor de informații. A publicat peste o sută de articole și câteva cărți și a obținut douăzeci de patente.

Dr. Jayant a susținut doctoratul în ingineria comunicațiilor electrice la Indian Institute of Science, din Bangalore, India. El a fost cercetător asociat la Stanford University timp de un an, înainte de a se alătura echipei de la Bell Laboratories, din Muray Hill, în 1968. Dr. Jayant a obținut câteva premii, printre care Premiul Memorial IEEE Browder J. Thompson și Premiul Publicației IEEE Donald G. Fink. Dr. Jayant este o figură proeminentă a IEEE și membru al Academiei Naționale Americane de Inginerie.

International Journal of High Speed Electronics and Systems, Vol. 8, No. 1 (1997) 13-68
© World Scientific Publishing Company

METODE CURENTE DE CODARE A VORBIRII

RICHARD V. COX
Speech Coding Research Department
AT&T Bell Laboratories
Murray Hill, NJ 07974, USA

Codarea digitală a vorbirii este folosită într-o mare varietate de aplicații de zi cu zi, pe care omul obișnuit le percepe ca normale, cum ar fi rețelele cablate, telefonia celulară și roboții telefonici. Acest articol trece în revistă unele dintre cele mai utilizate metode de codare a vorbirii. Aceasta include revederea atributelor de bază ale acestor codoare, metodele utilizate pentru codare și unele dintre cele mai importante standarde de codare a vorbirii. Se vor revedea, de asemenea, metodele utilizate pentru a realiza codoarele digitale de vorbire, în special acelea destinate implementării acestor codoare într-o manieră eficientă din punct de vedere economic.

1. Introducere

Trăim într-o eră a informației digitale. În fond, toate informațiile pe care le primim în fiecare zi au fost stocate sau transmise într-o formă digitală la un anumit moment, înainte ca noi să le recepționăm. Acest fapt a fost de mult timp valabil pentru anumite forme de informație, cum ar fi ziarele sau revistele, care se bazează în principal pe text. Până și imaginile au fost digitizate ca parte a procesului de producție. Transmiterea prin cablu a fotografiilor pentru ziare a folosit digitizarea din anii '20. Televiziunea și programele radio sunt încă distribuite prin metode analogice, dar în stadii intermediare ale distribuției, aceste programe sunt transmise digital, prin legături terestre sau prin satelit. Întreaga rețea telefonică din Statele Unite este 100% digitală, incluzând și comutatoarele digitale. Cele mai multe dintre serviciile telefonice locale sunt, de asemenea, direcționate prin comutatoare digitale, numai transmisia către locuințe și instituții rămânând analogică.

Dată fiind predominanța transmisiilor și memorărilor digitale, este ușor de înțeles de ce codarea digitală a vorbirii a devenit azi larg utilizată. Informațiile stocate în formă digitală comprimată, fac mult mai eficientă transmisia și stocarea și sunt mai ușor de procesat. Oricum, publicului larg îi este indiferent cât de mult din vorbirea pe care o aude este deja procesată prin codoare digitale.

Cum am mai arătat, transmisia digitală a vorbirii este folosită în rețelele telefonice. Codarea vorbirii folosită în rețeaua telefonică este pur și simplu cuantizarea eșantioanelor, unul după altul. Viteza de transmisie pentru cele mai multe apeluri este fixată la 64 kbps. Vorbirea este eșantionată cu 8000 Hz și se folosește o cuantizare logaritmică pe 8 biți pentru a reprezenta fiecare eșantion ca una dintre cele 256 de valori posibile. Convorbirile internaționale prin cabluri transoceanice sau satelit sunt adesea reduse la o rată de bit de 32 kbps pentru a mări capacitatea acestor echipamente relativ scumpe.

Transmisiunile digitale fără fir au început deja. În America de Nord, Europa și Japonia sunt deja operaționale sisteme de telefonie celulară cu rata de bit între 6,7 și 13 kbps pentru codoarele de vorbire. Există două standarde în competiție, pentru telefonia digitală fără fir. Acestea sunt DECT (*Digital European Cordless Telephone*) și CT2 (*Cordless Telephone 2*). Ambele folosesc codarea vorbirii cu 32 kbps. Corporația Inmarsat oferă curent servicii de telefonie prin satelit clienților săi aflați pe mare, în avioane sau în zone izolate. Alte consorții se formează pentru a asigura o acoperire și o capacitate mai bună prin sisteme de sateliți. Toate aceste sisteme vor folosi codoare cu rate de bit sub 10 kbps, mergând până la 2,4 kbps.

Telefonia cifrată există încă din timpul celui de-al doilea război mondial, bazată atunci pe primul vocoder[1] (de la voice = voce și coder = codor). Telefonia cifrată presupune conversia vorbirii în formă digitală, apoi criptarea (cifrarea) digitală și apoi transmisia. La recepție are loc decriptarea și conversia inversă, în semnal analogic. Adesea transmisia digitală este realizată prin folosirea unui modem. Standardele de telefonie cifrată curente prevăd rate de bit între 0,8 și 16 kbps. Telefoanele cifrate au fost folosite întâi pentru uz militar sau guvernamental. Recent, ele au început să fie folosite de corporații care sunt îngrijorate de posibilitatea supravegherii electronice a convorbirilor de către rivali.

Videotelefonia curentă este realizată prin transmisia digitală atât a vorbirii cât și a semnalelor video. Utilizatorii sunt atenți mai ales la calitatea semnalului video, dar nu ar putea să-și desfășoare conversația fără partea audio. Pentru videoconferințe este de fapt nevoie de o calitate audio mai bună decât în cazul telefoniei standard.

O utilizare în plină dezvoltare a codării vocii este transmiterea simultană a semnalelor de voce și de date. În acest caz, utilizatorii schimbă date (text, imagini, fax, sau orice altă formă de date digitale) în timp ce duc o conversație obișnuită. (Videofonul poate fi considerat o extensie a cazului general al transmisiei simultane de voce și date.)

Toate exemplele anterioare implică o conversație în timp real. Astăzi utilizăm codoare de vorbire pentru multe aplicații care ne fac viața mai ușoară. De exemplu, sistemul de poștă vocală sau roboții telefonici ne permit să lăsăm mesaje altor persoane. Acestea pot să-și asculte mesajul când doresc, eventual chiar de la capătul celălalt al lumii, prin comandă de la distanță. Aceeași metodă de stocare poate fi folosită pentru a difuza anunțuri către un număr mare de persoane.

O altă dezvoltare a utilizării codării vorbirii este multimedia. Unele forme de multimedia sunt transmise în direct (*live*) iar altele sunt stocate, însă cele mai multe implică o comunicare într-o singură direcție. De aceea o includem în aplicațiile de stocare. O aplicație tipică de multimedia ar putea implica vorbirea. Documentele multimedia pe calculator pot avea incluse fragmente de vorbire ca parte integrantă. Posibilitățile actuale permit utilizatorilor să realizeze adnotări vocale pe documentele stocate într-un calculator sau stație de lucru.

Această listă ar trebui să fie suficientă pentru a convinge majoritatea cititorilor că există azi multe aplicații care depind de codarea vorbirii. Mai mult, numărul acestor aplicații va crește mult în viitor. În timp ce lista creștea în trecut, tehnologia de codare a vorbirii a fost capabilă să țină pasul cu cerințele aplicațiilor. Sunt două motive principale pentru aceasta. Întâi, provocările tehnice în zona algoritmilor au primit răspuns prin algoritmi mai sofisticați de codare. Acești algoritmi se bazează din ce în ce mai mult pe modele de producere a vorbirii și pe modele perceptuale ale auzului

uman. În acest fel, s-au obținut grade superioare de compresie, păstrând o calitate adecvată. În al doilea rând, pe măsură ce complexitatea algoritmilor a crescut, avansarea tehnologiei de integrare pe scară largă (VLSI) a circuitelor integrate a păstrat pasul. Caii de bătaie l-au constituit circuitele de procesare digitală a semnalelor (DSP = Digital Signal Processing). De la prima lor producere în 1980[2] până în prezent, viteza lor de procesare și capacitatea memoriei interne au crescut foarte mult. Nu pare să existe nici un fel de moment de relaxare în tehnologia VLSI. În prezent, volumul de cercetare și dezvoltare în codarea vorbirii pare să fie angajat pe un drum fără sfârșit, deoarece din ce în ce mai multe organizații simt nevoia codării vorbirii în aplicațiile lor. Viitorul pare asemănător cu trecutul, în sensul unei cereri în creștere de codare a vorbirii pe măsura înmulțirii aplicațiilor, în paralel cu evoluția mijloacelor de implementare care, folosind VLSI, devin tot mai puțin costisitoare și tot mai performante.

Domeniul codării vorbirii este prea vast pentru a putea fi epuizat într-un singur articol. În acest articol ne vom concentra pe acele tehnici de codare a vorbirii care sunt cel mai adesea folosite în produse sau servicii. În continuare, acest articol este divizat în trei capitole. În primul capitol vom discuta bazele algoritmilor de codare a vorbirii. Pornind de la un nivel general, vom discuta atributele codării vorbirii, atribute pe care un inginer care realizează o aplicație trebuie să le ia în considerare atunci când definește cerințele codorului de vorbire. Apoi ne vom muta pe niveluri mai profunde și vom revedea tehnicile de bază de codare. Ne vom concentra în mod particular asupra codoarelor cu analiză-prin-sinteză și predicție liniară, deoarece această tehnică a stat la baza celor mai multe standarde de codare a vorbirii de înaltă calitate, create în ultima decadă. Al doilea mare capitol se referă la standardele de codare a vorbirii. Standardele sunt fundamentale pentru codarea vorbirii, astfel încât produse ale diferiților producători să poată comunica între ele. Vom trece întâi în revistă procesul de standardizare și apoi vom arunca o scurtă privire asupra principalelor organizații de standardizare și a activității lor. Vom include o descriere a standardelor curente și a celor propuse. Capitolul final se referă la procesul de realizare a codoarelor de vorbire. Cum transformăm un algoritm într-o componentă cu aplicație largă? Este un proces cu mulți pași, pe care îl vom revedea în aceast capitol final.

2. Bazele algoritmilor de codare a vorbirii

2.1. *Atributele codoarelor de vorbire*

Codoarele de vorbire au atribute care pot fi plasate în patru grupe: rata de bit, calitatea, complexitatea și întârzierea. Pentru o aplicație dată, unele dintre aceste atribute sunt predeterminate, în timp ce pentru celelalte există loc de negociere. De exemplu, canalul de comunicare poate limita rata de bit, sau considerații legate de costuri pot determina limitările de complexitate. Calitatea poate, de obicei, să fie crescută prin mărirea ratei de bit sau a complexității și, uneori, prin creșterea întârzierii. Cerințele, la fel ca și scopurile, trebuie să fie determinante pentru toate aceste atribute. În următoarele paragrafe vom discuta aceste atribute. Oricum, subiectul ce trebuie abordat în primul rând este lărgimea de bandă.

În această lucrare vom discuta, înainte de toate, *lărgimea de bandă a vorbirii în telefonie*. Acest termen este ușor nebulos. În rețeaua telefonică, vorbirea este întâi filtrată în banda 200-3200 Hz. În zona de jos, rețelele telefonice se confruntă cu faptul că semnalele

vorbirii sunt adesea alterate de alte zgomote, cum ar fi armonicile rețelei de curent electric (50 Hz sau 60 Hz). Trebuie, de asemenea, să fie eliminată orice componentă continuă (DC) din semnal. Prin urmare, un filtru trece sus este necesar în zona de jos a frecvențelor. În primele rețele telefonice, semnalizările de joasă frecvență se făceau între 50 și 200 Hz. Aceste motive arată de ce există un filtru trece sus și de ce este pus acolo unde este. Începutul benzii de trecere este, la rândul său, puțin nebulos. Uzual este localizat între 200 și 300 Hz, depinzând de sistem și de producătorul echipamentului. Filtrul trece jos este impus din teoria eșantionării. Vorbirea este eșantionată cu 8 kHz în rețeaua telefonică. Conform teoremei de eșantionare a lui Nyquist, banda semnalului reprezentativ ar fi de cel mult 4 kHz. Oricum, nu există un filtru trece jos ideal. Filtrele uzuale folosite în telefonie scad la 35 dB în jurul a 4 kHz, pentru a elimina produsele spectrale create de eșantionare. Chestiunea este însă unde începe curba filtrului trece jos. Punctul de 3 dB poate fi între 3 și 3,6 kHz. Pentru scopul nostru, vom spune că lărgimea de bandă a vorbirii în telefonie este de la 200 la 3200 Hz. Adesea este numită vorbirea de 3 kHz.

Mai există o a doua lărgime de bandă, care este de asemenea de interes. I se spune vorbirea de bandă largă. Rata de eșantionare este dublată la 16 kHz. Filtrul trece jos se presupune a începe să cadă la 7 kHz. În zona de jos, se presupune că vorbirea nu este alterată de zgomote de linie și numai componenta continuă se cere a fi filtrată. Prin urmare, frecvența de tăiere a filtrului trece sus este 50 Hz. Deci, când ne referim la vorbirea de bandă largă, înțelegem o bandă de 50-7000 Hz și o rată de eșantionare de 16 kHz. I se poate spune foarte bine vorbirea de 7 kHz.

2.1.1. Rata de bit

Rata de bit ne indică gradul de compresie pe care codorul o poate atinge. Vorbirea telefonică este eșantionată la 8 kHz și digitizată printr-o cuantizare logaritmică pe 8 biți, rezultând o rată de bit de 64 kbps. Pentru codoarele vorbirii de bandă telefonică, măsurăm gradul de compresie prin cât de mult poate fi scăzută rata de bit sub 64 kbps. Există curent standarde de rețele telefonice internaționale pentru codoare care operează de la 64 kbps până la 5,3 kbps. Codoarele vorbirii pentru standarde regionale celulare au gama de la 13 la 3,45 kbps și cele pentru telefonie cifrată au gama între 16 kbps și 800 bps. În fine, există codoare brevetate, care sunt utilizate, și care se întind pe toată banda.

Codoarele vorbirii nu trebuie să aibă o rată de bit constantă. Pot să apară avantaje dacă nu se transmite pe durata pauzelor din timpul conversației. De asemenea, nu este necesar să se păstreze o rată de bit constantă în momentele de început ale conversației.

În planificarea unei aplicații de codare a vorbirii, rata de bit este uzual un parametru care este determinat de alte considerente, cum ar fi banda disponibilă pentru transmisie, sau capacitatea de stocare a vorbirii. Adesea, rata de bit nu este fixată a priori, dar valoarea aproximativă este dată de la început. Dată fiind rata de bit, proiectantul trebuie să realizeze cea mai potrivită alegere pentru atributele care rămân, astfel încât să se potrivească cerințelor aplicației.

2.1.2. Întârzierea

Cele mai multe codoare de vorbire cu rată de bit scăzută sunt codoare de blocuri (*block coders*). Aceasta înseamnă că la un moment dat ele codează un bloc de vorbire, numit adeseori și cadru. În funcție de aplicație, întârzierea totală de codare a sistemului

este un multiplu al duratei cadrului. Întârzierea minimă a sistemului este în general de la 3 până la 4 ori mărimea cadrului. De exemplu, multe codoare de viteză scăzută au dimensiunea cadrului de 20 ms, rezultând o întârziere la transmisia într-un singur sens, de 60 până la 80 ms.

Întârzierea introdusă în comunicații de către codor este mai importantă pentru transmisii decât pentru aplicațiile de înmagazinare de date. În cazul conversațiilor în timp real, o întârziere de valoare mare a comunicației poate să impună un protocol stânjenitor pentru vorbitori. Valori mari ale întârzierilor, de 400 ms sau mai mult, fac imposibile conversațiile (de tip „full-duplex"). În plus, întârzierea accentuează problemele legate de ecouri și introduce necesitatea de a anula ecoul, atât în cadrul rețelei (pentru fenomene de ecou electric) cât și la nivelul echipamentelor terminale (pentru fenomene de ecou acustic). Dacă întârzierea crește, cresc în mod semnificativ și cerințele de eliminare a ecoului[3]. Cu alte cuvinte, apare un dezavantaj în plus în proiectarea sistemelor de transmisie o dată cu creșterea întârzierii.

Într-o teleconferință (comunicație în timp real între mai multe părți), se folosește un dispozitiv de interconectări în care fluxurile de biți provenind de la fiecare participant la conversație sunt decodate, sumate și apoi re-codate pentru transmisie. Aceasta dublează întârzierea pentru că au loc două codări și decodări complete. (Acesta are de asemenea un impact negativ asupra calității, pe care îl vom discuta mai târziu.) Prin contrast, pentru o aplicație cu stocare, cum ar fi poșta vocală, o întârzirere de o secundă va fi de neobservat pentru utilizator.

Întârzierea în codarea vorbirii poate fi împărțită astfel. Întâi, există o întârziere algoritmică. Unele codoare au un timp de *privire înainte* sau alte întârzieri inerente care se adaugă la lungimea cadrelor. Suma dintre lungimea unui cadru și alte întârzieri inerente constituie întârzierea algoritmică.

Codarea presupune calcule. Timpul necesar acestora este numit întârzierea de procesare. Ea depinde de viteza procesorului utilizat. Întârzierea de procesare poate fi mai mare sau mai mică decât lungimea cadrului. Dacă implementarea se face pentru o aplicație cu mai multe canale, se poate folosi un procesor de viteză mare și întârzierea de procesare pe canal poate fi numai o mică fracțiune din mărimea unui cadru. Dacă un singur cip implementează numai un codor, atunci este foarte probabil ca întârzierea de procesare să fie ceva mai mică decât mărimea unui cadru. Dacă ar fi fost mult mai mică, procesorul ar fi fost inactiv (*idle*) pentru o anumită perioadă de timp (implicând o implementare ineficientă). Dacă ar fi mai mare, procesorul nu ar mai avea timp suficient să proceseze cadrul. Dacă sunt necesare mai multe cipuri deoarece unul singur nu este suficient de rapid, atunci este de presupus că întârzierea de procesare este mai mare decât durata unui cadru.

Urmează apoi întârzierea de multiplexare. Adesea codorul de vorbire face parte dintr-un sistem mai larg. Deoarece este un codor de blocuri, biții lui nu apar continuu pentru transmisie. Un bloc de biți este asamblat de către codor și prezentat apoi pentru transmisie. La recepție întregul bloc de biți trebuie să fie reasamblat înainte de începerea decodării. Din aceste rațiuni apare o întârziere de multiplexare.

În sfârșit, avem de-a face și cu o întârziere de transmisie. Codorul poate folosi un canal de transmisie de mare viteză împreună cu alți utilizatori sau poate folosi un canal dedicat. Întârzierea de transmisie nu poate fi mai mare decât un cadru, dar poate fi mult mai mică. Ca element general de ghidare, considerăm în general că întârzierile de multiplexare și transmisie – în mod combinat – sunt egale cu mărimea unui cadru, atunci când aplicația presupune un canal dedicat.

Din punctul de vedere al proiectantului, întârzierea poate fi atât un parametru liber cât și fix pentru o aplicație particulară. Pentru aplicațiile de transmisie putem avea de-a face cu o mărime fixă a cadrului, sau cu o gamă de valori permise. În cazul aplicațiilor legate de stocarea de date, întârzierea este aproape totdeauna un parametru fără restricții și poate fi folosit fie pentru îmbunătățirea calității, fie pentru reducerea ratei de bit, fie pentru reducerea complexității.

2.1.3. *Complexitatea*

Gradul de complexitate este un factor determinant atât în costul cât și în puterea consumată a unui codor de vorbire. Costul este aproape întotdeauna un factor în selectarea unui codor de vorbire pentru o aplicație dată. O dată cu avântul comunicațiilor fără fir și portabile, puterea consumată a devenit și ea un factor important.

Cuantizatori scalari simpli, cum ar fi PCM liniar sau logaritmic, sunt necesari în orice sistem de codare și au complexitatea cea mai scăzută. Codoarele cu rata de bit mai mică au complexitate mai mare. Unele codoare au complexități similare, indiferent de ratele de bit pe care le au. De exemplu, codoarele PCM adaptative diferențial (ADPCM), standardizate de ITU-T[a] pentru a opera la rate de bit de 40, 32, 24 și 16 kbps au complexitate aproape identică, diferind numai prin cuantizatori. Prin urmare, în alegerea atributelor unui codor, poate fi analizat raportul între complexitate și rata de bit.

Codoarele de vorbire mai complexe sunt întâi testate pe un procesor gazdă, apoi implementate în circuite DSP și pot fi în final incluse în circuite specializate VLSI. Viteza și memoria RAM (memorie cu acces aleatoriu) sunt factorii care contribuie cel mai mult la complexitate. Cu cât cipul este mai rapid, sau cu cât are memoria mai mare, cu atât și costul este mai mare. Aceleași atribute influențează consumul de putere. Prin urmare, complexitatea este factorul determinant atât pentru cost cât și pentru puterea consumată. În general, un cuvânt de RAM, necesită pe cip o suprafață la fel de mare ca și 4-6 cuvinte de ROM (*read only memory*). În general există constrângeri asupra vitezei (MIPS = *Million Instructions Per Second* = milioane de instrucțiuni pe secundă), RAM-ului și ROM-ului pentru o aplicație dată, indiferent dacă implică transmisia sau stocarea. Cele mai multe codoare de vorbire sunt implementate pe circuite DSP cu virgulă fixă, astfel că o modalitate de a compara complexitatea codoarelor este de a măsura viteza și cerințele de memorie când sunt implementate eficient pe cipuri comerciale DSP cu virgulă fixă.

Cipurile DSP sunt disponibile atât pe 16 biți cu virgulă fixă cât și pe 32 de biți cu virgulă mobilă. Cele pe 16 biți sunt în general preferate pentru implementări dedicate ale codoarelor de vorbire, deoarece sunt de obicei mai ieftine și consumă mai puțin decât implementările bazate pe virgulă mobilă. (Implementările tipice de DSP cu virgulă mobilă necesită RAM sau ROM extern cipului, ceea ce mărește și costul și puterea consumată). Pe măsură ce tehnologia VLSI avansează, cantitatea de memorie și viteza (MIPS) pe care le poate avea un circuit cresc rapid. Primul cip DSP cu virgulă fixă produs de AT&T putea executa 1,25 MIPS și avea numai 128 B (B = baiți, octeți) de RAM și 1024 B (1KB) de ROM[2]. Cipurile DSP de azi, cu virgulă fixă, ale diferiților producători, sunt capabile de 40 MIPS și au 4 KB de RAM și 16 KB de ROM. Unul dintre cele mai avansate DSP este capabil de 70 MIPS și conține 6 KB (K octeți) de RAM și

[a] Mai multe detalii despre ITU-T (International Telecommunications Union Telecommunications Standardization Sector) pot fi găsite în capitolul 3 al acestui articol.

36 KB de ROM[4]. Circuitele DSP de azi consumă cu mult mai puțin și sunt relativ mai ieftine. Un codor cu implementare inițial scumpă va fi cu mult mai ieftin 5 ani mai târziu.

Un dezavantaj al cipurilor DSP cu virgulă fixă este că impun folosirea – pentru codarea vorbirii – a unui algoritm bazat pe o aritmetică pe 16 biți. Ca parte a unui proces de implementare, pentru fiecare variabilă trebuie selectată o reprezentare. Unele pot fi reprezentate într-un format fix, altele în blocuri cu virgulă variabilă, iar altele pot să necesite dublă precizie. Pe măsură ce tehnologia VLSI a avansat, cipurile DSP cu virgulă fixă conțin un set mai bogat de instrucțiuni pentru a gestiona manipulările de date care sunt cerute pentru implementarea unei reprezentări ca blocurile cu virgulă mobilă. Avantajul circuitelor DSP cu virgulă mobilă este că implementarea codoarelor de vorbire este mult mai rapidă. Precizia lor aritmetică este aproximativ identică cu a simulării limbajelor de nivel înalt, așa că pașii care determină reprezentarea fiecărei variabile și modul în care această reprezentare afectează performanțele, pot fi omiși. Pentru mai multe despre aceste probleme, a se vedea capitolul 4 al acestui articol.

2.1.4. Calitatea

Atributul calității are multe dimensiuni. În ultimă instanță, calitatea este determinată de modul în care vorbirea îi sună ascultătorului. Unii dintre factorii care afectează performanțele unui codor sunt măsura în care semnálul de intrare este curat sau zgomotos, dacă fluxul de biți a fost alterat de erori sau nu, dacă au avut loc codări multiple. Pentru fiecare dintre acestea ne punem întrebarea „cum se prezintă calitatea acestui codor în comparație cu performanțele obținute în condiții de referință (etalon) și în circumstanțe ideale sau similare?" Răspunsul determină calitatea codorului de vorbire.

ITU Speech Quality Experts Group (SQEG) este arbitrul suprem al ceea ce constituie calitatea vorbirii specifice rețelelor telefonice (*toll quality*). Recomandarea G.711 a CCITT pentru codarea vorbirii PCM cu 64 kbps a fost primul standard de codare a vorbirii. Pentru măsurarea calității vorbirii, SQEG a creat o unitate ipotetică numită unitate de cuantizare a distorsiunilor (QDU = *Quantization Distortion Unit*), care poate fi utilizată pentru a schița calitatea unei rețele. Distorsiunea introdusă de un codec (codor / decodor) tip G.711 este considerată un QDU. La vremea când a fost creat QDU, rețeaua telefonică tocmai devenea digitală și o convorbire internațională putea fi direcționată printr-o serie de legături de transmisiuni analogice și digitale. Astăzi este mult mai probabil ca o convorbire să fie transportată digital de la un capăt la celălalt.

Recomandarea G.726 pentru ADPCM cu 32 kbps a fost standardizată pentru a fi utilizată cu un codec G.711 la intrare și la ieșire. Această combinație se consideră a avea 3,5 QDU. Directivele SQEG de planificare a rețelelor impun un maxim de 14 QDU pentru o conexiune internațională de la un utilizator la altul și mai puțin de 4 QDU pentru o legătură locală.

Pentru vorbirea de 7 kHz, echivalentul calității din telefonia analogică (*toll quality*) este calitatea comentariului (*commentary quality*)[NT]. Din nou, este vorba de o calitate foarte înaltă, dar nu lipsită complet de distorsiuni. Calitatea comentariului poate fi folosită pentru teleconferințe și, de asemenea, pentru distribuția vorbirii prin legături de

[NT] Denumirea provine de la comentariile „telefonice" ale reporterilor retransmise din studiourile de radio sau televiziune.

televiziune sau radio. Cel mai uzual scenariu este introducerea unui comentariu al unui corespondent „de la fața locului" în transmisia din studio.

Nivelurile de calitate ale codoarelor de vorbire, cum este QDU, sunt determinate pe baza testelor de ascultare subiective. Ascultarea se face în cabine foarte silențioase, folosind terminale telefonice specifice, căști sau difuzoare. Materialul vorbit este prezentat ascultătorilor la niveluri de volum specificate și este inițial pregătit pentru a avea caracteristici predeterminate de frecvență.

Cel mai frecvent sunt utilizate teste de calitate, care determină nivelul absolut (ACR = *Absolute Category Rating*) sau teste de calitate, care determină nivelul de degradare a calității (DCR = *Degradation Category Rating*). În testul ACR, subiecții ascultă o pereche de propoziții procesate de un anumit sistem, după care dau unul din următoarele calificative: excelent, bine, suficient, slab, rău. În testul DCR, subiecții ascultă două perechi de propoziții. Cel mai adesea prima este materialul sursă, original, iar a doua pereche de propoziții este rezultatul procesării, fie de către codorul supus testului, fie în condiții de referință. Subiecții sunt apoi rugați să aprecieze degradarea perceptibilă a celei de-a doua perechi de propoziții, comparativ cu prima. Calificativele posibile sunt: imperceptibil, perceptibil dar nederanjant, ușor deranjant, deranjant dar suportabil, foarte deranjant (insuportabil). Folosirea testului DRC de către SQEG este relativ nouă și s-au experimentat recent teste folosind un set nou de calificative care evită folosirea cuvântului „deranjant". Acestea se referă la al doilea set de propoziții în comparație cu primul și sunt: la fel sau mai bine, puțin mai rău, o înrăutățire moderată, mai rău, mult mai rău. Deoarece subiectul trebuie să compare semnalele în toate condițiile posibile, testul DCR ia de două ori mai mult timp pentru a fi realizat decât testul ACR, pentru a obține același volum de informație.

Datele de la aceste teste pot fi analizate în multe feluri. Calea cea mai simplă este de a aloca un echivalent numeric fiecărui tip de răspuns, dând valoarea 5 pentru nivelul cel mai bun, 4 pentru următorul și așa mai departe până la 1 pentru nivelul cel mai rău și apoi calculând media pentru toate condițiile în care s-a făcut testul. Pentru testul ACR, acest rezultat este numit scorul mediu de apreciere (MOS = *Mean Opinion Score*), iar pentru testul DCR, este numit scorul mediu de degradare (DMOS = *Degradation Mean Opinion Score*). Metode mai sofisticate de analiză (ANOVA) pot fi folosite pentru a determina dacă diferența între condiții trebuie considerată a fi statistic semnificativă. Altă metodă este de a ridica histograma brută a datelor. Furnizorii de servicii de rețea au descoperit că există o corelație între numărul de plângeri asupra calității rețelei pentru anumite condiții și procentul de voturi „slab" sau „rău" din testul ACR pentru acele condiții. Pentru a determina cu cât sunt mai rele unele condiții decât altele, pare a fi mai bine să luăm în considerație creșterea procentului de voturi „slab" sau „rău" decât să vedem care este diferența între mediile MOS.

În timp ce ITU s-a concentrat să mențină calitatea telefonică a vorbirii (*toll quality*) în rețelele cablate, alte foruri de standardizare au creat standarde de calitate mai redusă. Telefonia digitală celulară a fost standardizată de organisme regionale din Europa, America de Nord și Japonia. Prima generație de standarde pentru telefonia celulară digitală au asigurat canale clare de vorbire, care sunt greu de comparat. Calitatea vorbirii nu este totuși la fel de bună ca a codorului original de 64 kbps sau ca a codorului G.726 de 32 kbps. Din nefericire, adesea ele nu sunt la fel de clare nici ca telefonia celulară analogică, ceea ce a creat o publicitate negativă pentru telefonia digitală celulară, mai ales în Statele Unite. Organismele de standardizare din Europa și America de Nord

vizează adesea aceste neajunsuri prin programe care să amelioreze calitatea noilor codoare de vorbire.

Codoarele de vorbire cu rate de bit joase au fost create pentru alte aplicații. Vorbirea este încă foarte inteligibilă, dar există destule distorsiuni adiționale care duc la obosirea ascultătorului în timpul unei conversații prelungite. Calitatea slabă este, de asemenea, măsurabilă prin rezultatele testelor subiective. Codoarele cu ratele de bit cele mai scăzute, cum ar fi cele folosite pentru telefonia protejată (cifrată), sunt adesea testate pentru determinarea gradului lor de inteligibilitate folosind testul de diagnosticare a rimelor (DRT = *Diagnosis Rhyme Test*). Este acceptată o calitate mai slabă a acestor codoare datorită ratei mici de bit, dar este esențial să nu fie atât de slabe încât să fie compromisă inteligibilitatea vorbirii.

Performanțele codoarelor cu rate de bit reduse au de suferit mai ales când intrarea conține alte sunete decât vorbirea, cum ar fi zgomotul de fond sau muzică (fie ca semnal primar sau ca fundal sonor). Pentru a obține compresia sunt utilizate modele de producere a vorbirii. (Mai multe despre aceasta în capitolul 2.2.) Aceste modele nu sunt atât de solide încât să asigure modelarea vorbirii în combinație cu alte sunete. Când aceasta se întâmplă, celălalt semnal este reprodus de codorul de vorbire ca niște cârâieli, ciripituri, scârțâieli, lătrături și alte astfel de efecte sonore ciudate. Aceste artefacte pot provoca inconsecvențe sau pot face codorul de neutilizat.

Performanțele unui codor de vorbire cu rată de bit joasă pot fi adesea maximizate pentru aplicația pentru care a fost destinat. Multe codoare de vorbire au fost gândite pentru folosirea în telefoane mobile. Pentru unele aplicații ca videofonul, codorul de vorbire poate fi utilizat cu un microfon de voce (independent). Setarea parametrilor pentru obținerea unui sunet optim în telefon, poate să nu fie la fel de bună pentru un difuzor de putere. În plus, caracteristica spectrală de intrare pentru un microfon de telefon, diferă de cea a unui microfon de voce. Dacă un codor de vorbire poate fi folosit în ambele situații, atunci trebuie testat pentru ambele moduri și trebuie setat astfel încât să se obțină cel mai bun compromis.

Sunt multe alte dimensiuni ale calității în afara celor care țin de canalele nezgomotoase. Sensibilitatea față de erorile de bit este un alt aspect al calității. Pentru canalele radio din telefonia digitală celulară, se face loc unor biți adiționali care sunt folosiți la codarea canalului, pentru protejarea biților care transportă informația. Nu toți biții unui codor de vorbire sunt la fel de sensibili. Este o practică uzuală să se stabilească două sau trei clase, cu cei mai sensibili biți primind protecția cea mai mare, iar cei mai puțin sensibili biți rămânând neprotejați. Pentru unele aplicații cu rate de bit joase, cum ar fi telefoanele protejate prin modemuri de 2,4 – 4,8 kbps, ar fi rezonabil să ne așteptăm ca distribuția erorilor de bit să fie aleatoare. Pentru modemuri cu rate mari, semnalele de eroare sunt mai probabile în salve. În acest caz, codorul necesită un mecanism de recuperare a unui întreg cadru pierdut. Aceasta se va numi ascunderea pierderii unui cadru. Este un alt aspect al calității, care trebuie judecat pentru codoarele celulare.

În scopul conservării lărgimii de bandă, se folosesc uneori, alături de codoare de vorbire, detectoare de activitate vocală. Pe parcursul intervalelor de timp fără vorbire, fluxul de biți al codorului devine discontinuu. La recepție se injectează în aceste perioade un „zgomot de confort" (*comfort noise*) pentru a simula zgomotul de fundal la codor. Această metodă este utilizată pentru anumite sisteme celulare și, de asemenea, în sistemele

de interpolare digitală a vorbirii (DSI = *Digital Speech Interpolation*) pentru a mări numărul efectiv de canale sau circuite. Multe legături telefonice internaționale realizate prin cabluri sub-oceanice sau sateliți folosesc acum sisteme tip DSI.

Există un impact asupra calității la utilizarea acestor tehnici. Testări cu caracter subiectiv pot determina nivelul de degradare. Utilizarea metodelor DSI poate prezenta două probleme. Dacă detectorul de activitate vocală este lent în detectarea vorbirii, sunetul de început al vorbirii se pierde. Numim aceasta „decuparea începutului" (*front-end clipping*). Problema decupării începutului stă în sarcina detectorului de activitate vocală. A doua problemă apare când există un zgomot acustic pe fundalul vorbirii. Zgomotul va interfera cu semnalul vorbirii, dar va fi eliminat în timpul absenței vorbirii. Aceasta are asupra ascultătorului un efect deconcertant și poate avea impact asupra inteligibilității. În funcție de caracterul zgomotelor din fundal, se pot introduce porțiuni de „zgomot de confort" pe perioada absenței vorbirii, astfel încât ascultătorul să audă un zgomot de fond și pe durata perioadei când nu se vorbește.

2.2. *Predicția liniară bazată pe modelele de producere a vorbirii*

În decursul anilor, tehnicile de codare a vorbirii cu cel mai mare succes s-au bazat pe codarea cu predicție liniară (LPC = *Linear Prediction Coding*)[b]. Elementul esențial al LPC este filtrul cu predicție liniară. Acesta este un filtru care prezice valoarea următorului eșantion, bazându-se pe o combinație liniară a eșantioanelor precedente. Dorim să demonstrăm în acest capitol de ce LPC este un model folositor pentru codarea vorbirii și să oferim o scurtă introducere în matematica LPC.

Vorbirea umană este produsă în tractul vocal printr-o combinație între coardele vocale din glotă în interacțiune cu articulatorii tractului vocal. Articulatorii sunt limba, buzele și vălul palatin. Dacă ei se combină pentru a crea o constricție, se produce un sunet ca un zgomotot, cum sunt „s", „ș", „z". Aceasta este determinată de turbulențele de aer provocate de constricție. Dacă buzele sunt închise și apoi deschise, se produce un sunet exploziv, cum este „p". Vălul palatin închide și deschide calea aerului prin tractul nazal. Sunetele nazale includ pe „m" și „n". Coardele vocale se pot deschide și închide periodic pentru a produce sunete ale vorbirii, cum ar fi vocalele. Sau pot rămâne deschise în timpul producerii unor sunete ne-vocalice. Figura 1(a) arată tractul vocal și modul cum se pot mișca articulatorii pentru a schimba sunetele produse.[5] Figura 1(b) este o diagramă a modelului aproximativ al lui Flanagan și Ishizaka de producere a vorbirii.[6] El include și presiunea plămânilor, traheea, bronhiile și coardele vocale.

Tractul vocal poate fi aproximat cu un tub de diametru variabil. Forma tubului dă naștere frecvențelor rezonante. În procesul vorbirii ne referim la aceste rezonanțe ca la formate date. Dacă examinăm spectrograma vorbirii de bandă largă, ca în figura 2, putem vedea formatele de frecvențe ale vorbirii și cum se schimbă în timp. În figură, frecvența este pe axa verticală, iar timpul este pe axa orizontală. Durata este aceea a rostirii de către un bărbat a propoziției „the small pup gnawed a hole in the sock". Cantitatea de frecvențe la un anumit moment este modulația între alb (conținut mic) și negru (conținut relativ mare). Traiectoriile întunecate din figură indică traiectoria în timp

[b] Numărul standardelor de codare a vorbirii, care conțin filtre de sinteză cu predicție liniară este mai mare decât numărul tuturor celorlalte standarde la un loc.

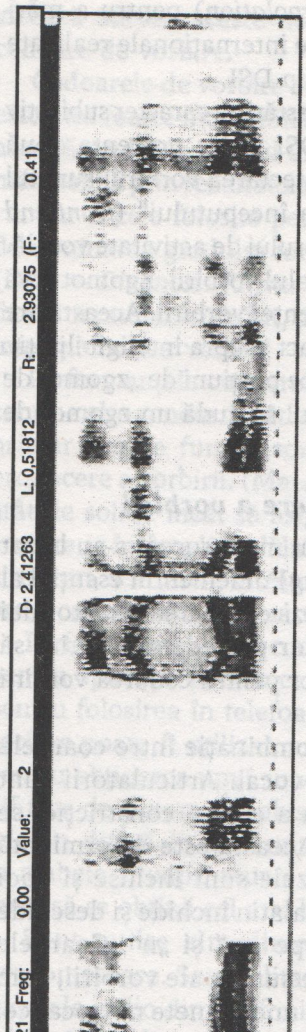


Figura 2. Spectrograma de bandă largă a propoziției „the small pup gnawed a hole in the sock"

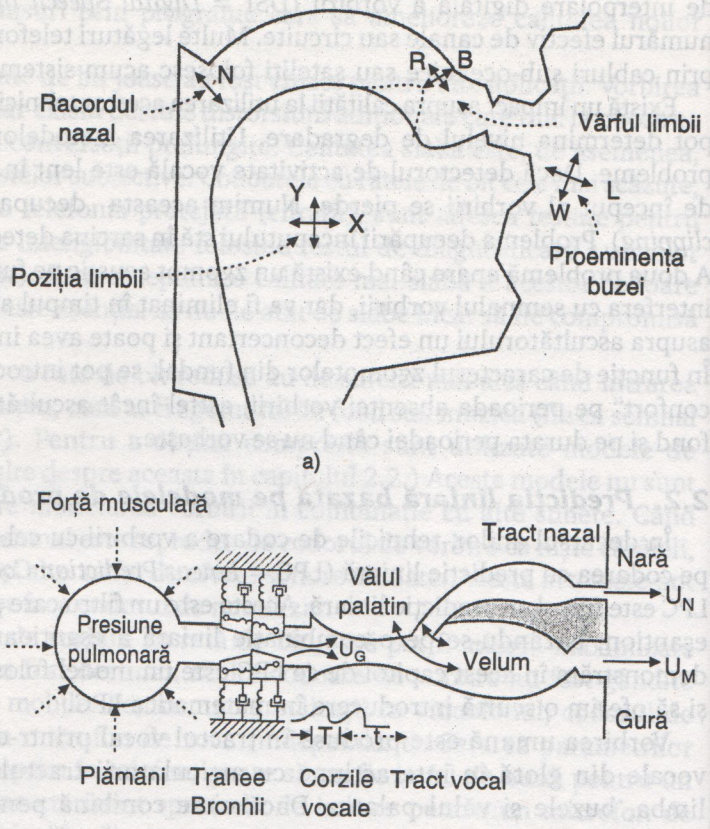


Figura 1. (a) Articulatorii tractului vocal, (b) modelul mecanic al tractului vocal

a diferitelor formate. Tipic există între trei și cinci formate asociate fiecărui sunet al vorbirii umane. Poziția relativă a acestor formate face distincția între diferitele foneme. (Și succesiunea fonemelor este cea care exprimă informația în vorbire). Cât de fidel sunt reproduse aceste foneme de codorul de vorbire, iată ceea ce determină inteligibilitatea codorului de vorbire.

LPC a derivat dintr-o aproximare matematică a tractului vocal, reprezentat ca un tub de diametru variabil. Să presupunem că tubul ar fi modelat ca un grup de tuburi de lungimi egale și diametre diferite, ca în figura 3. Acest model poate fi descris exact matematic printr-un set de ecuații. La îmbinarea între cilindri vor avea loc unele reflexii ale undelor sonore și unele propagări înainte. Aceste reflexii sunt modelate prin *coeficienții de reflexie*, iar combinația pentru întregul tub determină frecvențele lui de rezonanță.

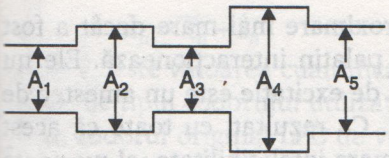


Figura 3. Tub acustic modelat de LPC

Figura 4 este o diagramă descriind modelul matematic. Acesta este un model pentru o rețea de filtre cu răspuns finit la un impuls. Este de fapt filtrul invers, arătând cum poate fi calculat semnalul de excitație, e_n, din semnalul vorbirii, s_n. Semnalul b_n reprezintă componenta propagată înapoi, iar coeficienții de reflexie sunt dați de k_i. Rețeaua de filtre poate fi transformată în obișnuitul filtru FIR (= *Finite Impulse Response*). Descrierea matematică a acestei transformări nu este foarte importantă pentru noi în acest punct, dar este de notat că toate aceste modelări matematice au fost folosite inițial pentru a descrie liniile de transmisie analogice. Prin urmare, proprietățile variate ale acestor sisteme sunt bine înțelese în literatura de specialitate. Inversul unui filtru FIR este un filtru IIR (*Infinite Impulse Response* = cu răspuns infinit la impuls) fără zerouri. În acest caz, el este cunoscut ca filtrul de sinteză LPC. Un filtru de sinteză LPC de ordinul 5 este arătat în figura 5.

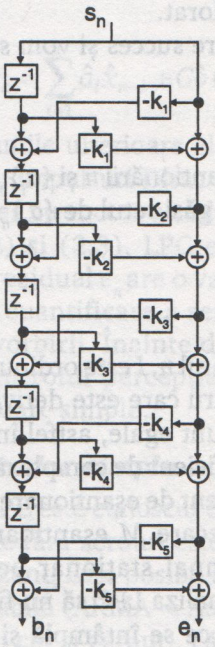


Figura 4. Răspunsul rețelei de filtre la impulsuri finite

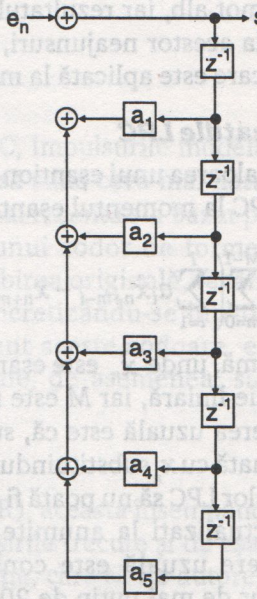


Figura 5. Filtrul de sinteză de ordinul 5 LPC în forma directă

Filtrul IIR fără zerouri pe care tocmai l-am descris, este numai o aproximare matematică a tractului vocal. Ca o consecință a imperfecțiunilor modelului, el nu poate produce sunete de vorbire cu adevărat naturale, la rate de bit foarte joase.

Un tract vocal real are pereți moi, care au pierderi acustice. Modelul descrie un sistem cu pereți „tari", în care nu există nici o disipare a energiei acustice pe pereți. În modelul original LPC, semnalul de excitație a fost decuplat de la filtrul IIR. Acțiunile vălului palatin au fost modelate ca producând fie un tren periodic de impulsuri, fie un

zgomot alb. Aceasta a fost pentru vălul palatin o aproximare mai mare decât a fost filtrul IIR pentru tractul vocal. Tractul vocal și vălul palatin interacționează. Ele nu sunt decuplate ca în modelul LPC. Și adesea semnalul de excitație este un amestec de componente, unele periodice și altele de tip zgomot. Ca rezultat, cu toate că acest model poate fi folosit pentru a produce vorbire cu o mare inteligibilitate, el nu poate produce totuși vorbire cu sunet natural.

De asemenea, se poate ușor aprecia că nu toate sunetele pot fi modelate prin cuplarea unui generator de trenuri de impulsuri periodice sau de zgomot alb cu un filtru IIR. De exemplu, muzica este un semnal mult mai complex. Un sunet individual este probabil să aibă un număr mare de armonici pare. Sunetul unei coarde este o combinație de câteva tonuri care sună plăcut urechii. Modelul de mai sus poate reproduce un singur ton (dacă frecvența fundamentală a tonului este în plaja acustică umană) dar nu poate modela sunetul unei corzi. Un alt exemplu este acela al sunetului unei taste telefonice, care este o combinație a două tonuri ne-legate armonic. Filtrul LPC ar putea să realizeze un model aproximativ al acestor două „rezonanțe", dar semnalul de excitație ar trebui să fie zgomot alb, iar rezultatul ar suna ca un zgomot colorat.

În ciuda acestor neajunsuri, LPC este o tehnică de mare succes și vom studia acum modul în care este aplicată la modelarea vorbirii.

2.3. *Ecuațiile LPC*

Fie x_n valoarea unui eșantion de vorbire la momentul eșantionării n și $\{a_i\}_n$ coeficienții filtrului LPC la momentul eșantionării n. Scopul este de a găsi setul de $\{a_i\}_n$ astfel încât

$$\sum_{m=0}^{M-1}\left(\sum_{i=1}^{I} a_i \hat{x}_{n+m-i} - x_{n+m}\right)^2 \tag{1}$$

să fie minimă, unde $\hat{x}_n$ este eșantionul cuantizat la momentul n, I este ordinul modelului de predicție liniară, iar M este mărimea intervalului pentru care este derivat filtrul LP. Presupunerea uzuală este că, statistic vorbind, $\hat{x}_n$ și x_n sunt egale, astfel încât analiza este efectuată cu x_n substituindu-l pe $\hat{x}_n$. Procedura este suficient de complexă ca valorile coeficienților LPC să nu poată fi actualizate la fiecare moment de eșantionare. În schimb, ei sunt actualizați la anumite intervale regulate (la fiecare M eșantioane). O altă presupunere uzuală este considerarea vorbirii ca semnal staționar pe parcursul intervalelor de mai puțin de 20 ms, ceea ce permite ca analiza LPC să nu fie efectuată mai des de atât. În realitate aceasta nu este exact ceea ce se întâmplă și actualizări mult mai frecvente (ale coeficienților) tind să îmbunătățească performanța. Odată ce avem coeficienții cuantizați LPC $\{\hat{a}_i\}$ pentru un interval, atunci eșantionul previzionat, $\tilde{x}_n$, este dat de

$$\tilde{x}_n = \sum_{i=1}^{I} \hat{a}_i \hat{x}_{n-i} \tag{2.1}$$

semnalul de eroare este

$$e_n = x_n - \tilde{x}_n \tag{2.2}$$

iar valoarea eșantionului de ieșire este

$$\hat{x}_n = \tilde{x}_n + \hat{e}_n \tag{2.3}$$

unde $\hat{e}_n$ este valoarea cuantificată a lui e_n. Modul în care este cuantificat e_n este determinat de tipul codorului utilizat.

În codorul original LPC de voce (*vocoder*) această paradigmă nu a fost utilizată. În schimb, vorbirea a fost declarată sonoră (cu voce, *voiced*) sau mută (fără voce, *unvoiced*).

Dacă vorbirea era considerată mută, atunci

$$\hat{x}_n = \sum_{i=1}^{I} \hat{a}_i \hat{x}_{n-i} + Gw_n \tag{3}$$

unde w_n este un zgomot alb gaussian iar G este câștigul LPC. Scopul era de a produce un semnal de ieșire $\hat{x}_n$ care are aceeași estimare statistică de ordinul 2 ca și x_n, în speranța că va suna la fel. Dacă vorbirea era sonoră, atunci semnalul de excitație era un tren periodic de impulsuri cu perioada P, astfel încât

$$u_n = \delta(n \bmod P) \tag{4.1}$$

și

$$\hat{x}_n = \sum_{i=1}^{I} \hat{a}_i \hat{x}_{n-i} + G\delta(n \bmod P) \tag{4.2}$$

În versiunile ulterioare ale codorului de vorbire LPC, impulsurile modelate au fost înlocuite cu impulsuri Dirac pure, făcând ca semnalul să sune ceva mai natural.

Utilizarea cu succes a codoarelor LPC în perioada mai recentă s-a bazat pe ecuațiile (2.1), (2.2) și (2.3). LPC este utilizată ca parte a unui codor de forme de undă. Semnalul rezidual e_n are o variație mai mică decât vorbirea originală, făcând posibilă o mai fină cuantificare a semnalului de eroare și concretizându-se într-o mai bună calitate a vorbirii. Înainte de a descrie mai în amănunt aceste codoare, este nevoie să discutăm rolul percepției în auzul uman și trebuie, de asemenea, să descriem câteva codoare simple.

2.4. *Un codor simplu – ADPCM*

Filtrul LPC este cunoscut ca un filtru autoregresiv (AR), aceasta însemnând că ieșirea filtrelor sale fără zerouri este dependentă numai de ieșirile trecute și de eșantionul de intrare prezent. Este posibil să formăm filtre de predicție, care să fie autoregresive, cu medie variabilă (ARMA = *AutoRegressive Moving Average*). Aceasta înseamnă că filtrele au atât poli cât și zerouri, că ieșirile lor sunt o funcție directă atât de ieșiri trecute cât și de intrări trecute. Am stabilit deja că polii corespund unor vârfuri spectrale cu formate cunoscute. Zerourile corespund minimelor (văilor) spectrale. (Zerourile pot fi folosite pentru a adăuga o fază mai corectă predictorului, în opoziție cu predicția minimă de fază a unui filtru fără zerouri. Semnificația acestui lucru este că în timpul vorbirii sonore, semnalul de excitație va fi și mai asemănător impulsurilor. Teoretic, aceasta ar trebui să îmbunătățească predicția.)

În 1984, CCITT[c] a standardizat un codor cu modulația diferențială adaptivă a impulsurilor (ADPCM = *Adaptive Differential Pulse Code Modulation*) pe 32 kbps, cu

[c] CCITT este Comitetul Consultativ Internațional pentru Telefonie și Telegrafie, predecesorul actualului ITU-T. Reorganizarea a avut loc în 1993. Mai multe detalii pot fi găsite în capitolul 3.

2 poli și 6 zerouri în predictorul adaptiv. Filtrul ARMA face o actualizare eșantion-cu-eșantion, folosing algoritmul gradientului mediei pătratice minime, cu o condiție la limită strictă pentru a asigura stabilitatea. El are, de asemenea, un cuantizator adaptiv cu 15 niveluri (4 biți), care își reglează dimensiunea pasului în funcție de nivelul semnalului. Codorul a fost proiectat să lase să treacă toate tonurile de semnalizare și semnalele de modem până la 4800 bps inclusiv. Pentru aceste tipuri de semnale, cuantizatorul filtrului adaptiv are pașii (treptele) blocați pe un nivel fix.

Ecuațiile pentru predictor sunt

$$\tilde{x}_n = a_1\hat{x}_{n-1} + a_2\hat{x}_{n-2} + \sum_{i=1}^{6} b_i\hat{e}_{n-i} \tag{5.1}$$

$$e_n = x_n - \tilde{x}_n \tag{5.2}$$

și

$$\hat{x}_n = \tilde{x}_n + \hat{e}_n \tag{5.3}$$

De remarcat că $\{a_i\}$ și $\{b_i\}$ nu sunt cuantizați. Decodorul are acces la aceeași informație ca și codorul și astfel coeficienții predictorului pot fi refăcuți la decodor. Numim aceasta adaptare recursivă. Ea mai înseamnă că nu este nevoie să cuantificăm și să transmitem coeficienții prin canalul de transmisie. Ecuațiile exacte pentru refacerea lor și condițiile de stabilitate pot fi găsite în [7].

Codorul ADPCM tip ITU-T G.726 servește ca un bun exemplu pentru introducerea codoarelor cu predicție liniară bazate pe analiză-prin-sinteză. Un filtru cu predicție liniară prezice valoarea următorului eșantion de vorbire. Valoarea prezisă este scăzută din valoarea măsurată. Cuantizatorul selectează valoarea $\hat{e}_n$ care este cea mai apropiată de e_n. O metodă alternativă de selecție a valorii cuantizate celei mai bune ar fi de a încerca să minimizăm diferența între x_n și semnalul de ieșire, $\hat{x}_n$. Se poate arăta că pentru cuantizări scalare diferențiale, aceeași valoare este optimală pentru ambele metode. Întrucât prima metodă implică mai puțini pași, este implementată pentru ADPCM G.726. Oricum, pot fi obiecții la afirmația că ADPCM G.726 ar fi un codor cu predicție liniară bazată pe analiză-prin-sinteză. Vom reveni la această idee a analizei-prin-sinteză când vom încerca să cuantizăm dintr-o dată un bloc de eșantioane de excitație.

Înainte de a părăsi ADPCM, vom mai spune că un filtru AR poate fi utilizat ca predictor. Coeficienții LPC pot fi cuantizați și transmiși pe canal. Transmisia explicită a coeficienților filtrului LPC îl face un filtru adaptiv-direct (nerecursiv). Un astfel de codor a fost creat și a fost numit codor adaptiv cu predicție (APC = *Adaptive Predictive Coding*).[8] Folosind tehnici speciale de cuantizare, care limitează numărul de eșantioane de excitație diferite de zero, rata de bit a putut fi scăzută până la 16 kbps.

2.5. *Modele ale percepției umane folosite în codarea vorbirii*

Cuantizatorul din codorul ADPCM minimiza eroarea pătratică medie între intrare și semnalul codat de la ieșire. Oricum, urechea noastră are o plajă de dinamică limitată care depinde atât de frecvență, cât și de densitatea spectrală, adică de semnal. Amintiți-vă că filtrul tipic trece bandă pentru telefonie are o bandă de oprire limitată la numai 35 dB. De asemenea, caracteristicile definite de legile μ și A specificate de Recomandarea G.711 a CCITT, conduc la un raport semnal / zgomot de cuantizare de aproximativ 35 dB.

Să fie o coincidență? Desigur că nu. Dacă un semnal își păstrează raportul S/N mai mare sau egal cu 35 dB, atunci cei mai mulți dintre oameni vor percepe un zgomot foarte mic sau chiar de loc, pentru banda telefonică.

Acest fenomen de mascare a zgomotului a fost cercetat printr-o serie de experimente în care ascultătorului i s-au reprodus un ton și un zgomot de bandă îngustă centrat pe tonul respectiv.[9,10] Au fost modificate raportul S/N și banda zgomotului. S-au obținut două rezultate importante. Dacă raportul S/N este suficient de mare, ascultătorul nu aude zgomotul. Sunetul a mascat zgomotul. Pe măsură ce banda zgomotului se mărește sau se centrează mai departe de ton, raportul S/N trebuie să fie mai mare pentru a masca zgomotul. Experimente similare s-au realizat pentru a determina zgomotul care să mascheze tonurile. Împreună, toate aceste experimente au condus la ideea de „bandă critică". În interiorul unei benzi critice, un semnal poate masca zgomotul, cu condiția ca raportul semnal / zgomot să fie suficient de mare. În afara acestei benzi, proprietățile de mascare sunt diminuate masiv. Pe măsură ce ne deplasăm de la frecvențe joase spre înalte, banda critică se mărește.

Au trecut cam 50 de ani de când acest important fenomen auditiv a fost descoperit, până când a fost aplicat pentru prima dată la codarea vorbirii. (Desigur, chiar codarea vorbirii a trebuit să fie inventată în această perioadă!) A fost prima utilizare a conceptului de filtru perceptual ponderat în codarea vorbirii[11]. Mai recent, aceste proprietăți de mascare au fost aplicate și la codarea audio de înaltă calitate[12]. (Presupunând că proprietăți asemănătoare pot fi constatate și la ochi, ele au fost măsurate empiric și în domeniul video și apoi aplicate la codarea imaginilor statice și dinamice, cu același succes)[13]. În acest articol vom discuta numai aplicațiile acestor proprietăți în domeniul codării vorbirii.

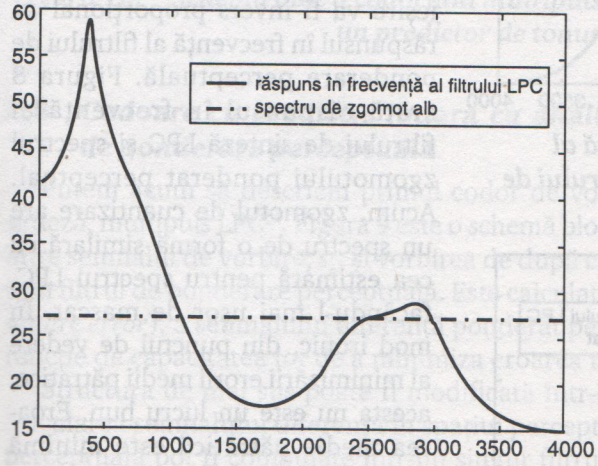


Figura 6. Răspunsul în frecvență al filtrului LPC (linia continuă) și spectrul zgomotului alb (linia punctată)

Să presupunem că spectrul unei vorbiri liniștite este cel dat de răspunsul în frecvență al filtrului LPC, așa cum este ilustrat în figura 6. Conceptual, proprietățile de mascare ne spun că putem permite un zgomot mai mare în regiunile dominante și în vecinătatea lor și că zgomotul va fi mai audibil în „văile" spectrale. Dacă folosim un codor care produce o caracteristică de zgomot alb, atunci spectrul zgomotului este plat și este de asemenea arătat în figura 6. Zgomotul va fi probabil auzit peste tot, mai puțin în prima regiune dominantă, centrată pe 400 Hz.

În referința bibliografică [14], autorul a judecat că, din moment ce informația filtrului LPC este disponibilă și constituie cea mai bună estimare a spectrului vorbirii, ar trebui folosită pentru a pondera sau a contura forma zgomotului de cuantizare a semnalului de excitație. Zgomotul de cuantizare al codorului vorbirii ar trebui minimizat într-un „spațiu perceptual". Prin urmare, a numit acest lucru „filtru de ponderare perceptuală".

Fie

$$A(z)=\sum_{i=1}^{I} a_i z^{-i} \qquad (6)$$

atunci filtrul de sinteză LPC este caracterizat de

$$\frac{1}{1-A(z)} \qquad (7)$$

Filtrul de ponderare perceptuală este caracterizat de

$$W(z)=\frac{1-A(z/\gamma_1)}{1-A(z/\gamma_2)},\quad 0<\gamma_2<\gamma_1\leq 1 \qquad (8)$$

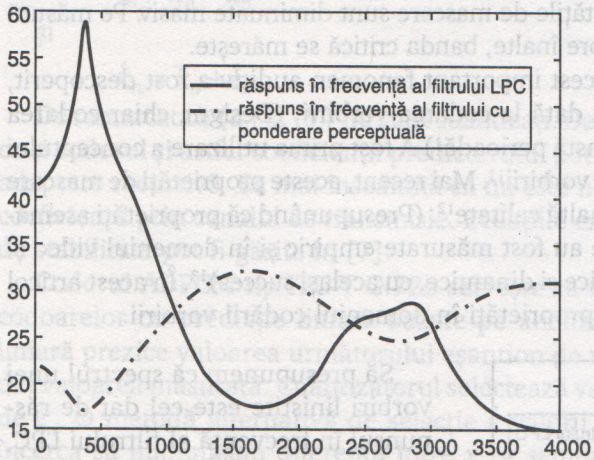


Figura 7. Răspunsul în frecvență al filtrului LPC (linia continuă) și al filtrului de ponderare perceptuală

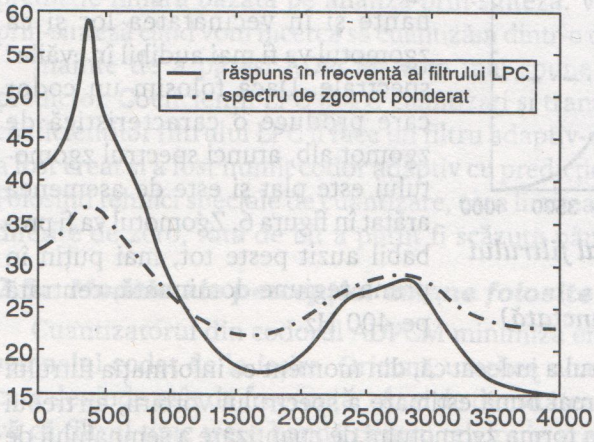


Figura 8. Răspunsul în frecvență al filtrului LPC (linia continuă) și spectrul ponderat al zgomotului (linia punctată)

Efectul acestui filtru este ilustrat în figurile 7 și 8. Figura 7 arată răspunsul în frecvență al filtrului de ponderare perceptuală suprapus peste răspunsul în frecvență al filtrului LPC. Filtrul de ponderare perceptuală atenuează importanța zgomotului în regiunile dominante și mărește importanța acestuia în văile spectrale. Spectrul zgomotului la ieșire va fi invers proporțional cu răspunsul în frecvență al filtrului de ponderare perceptuală. Figura 8 arată răspunsul în frecvență al filtrului de sinteză LPC și spectrul zgomotului ponderat perceptual. Acum, zgomotul de cuantizare are un spectru de o formă similară cu cea estimată pentru spectrul LPC, făcându-l mai ușor de mascat. În mod ironic, din punctul de vedere al minimizării erorii medii pătratice, acesta nu este un lucru bun. Eroarea medie pătratică este minimă atunci când zgomotul este alb. Prin urmare, modelarea sau colorarea zgomotului înseamnă că va fi mai mult, nu mai puțin zgomot de cuantizare. Cu toate acestea, datorită proprietăților de mascare ale urechii, zgomotul va fi mai puțin perceptibil.

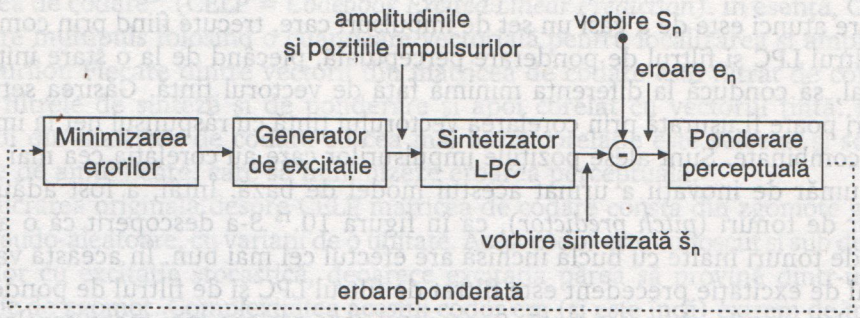


Figura 9. Schema bloc a codorului de vorbire multipuls cu predicție liniară

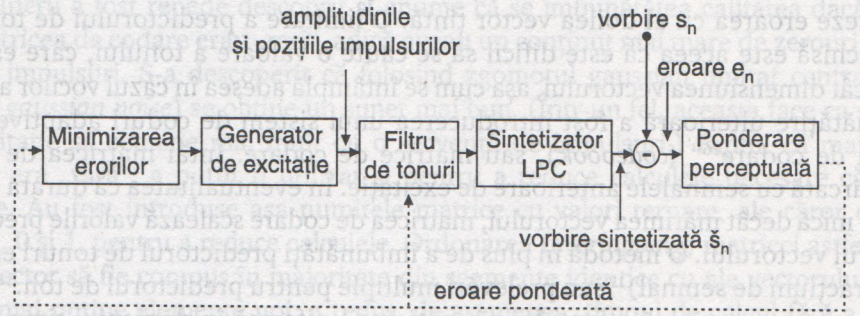


Figura 10. Schema bloc a codorului multipuls cu predicție liniară având adăugat un predictor de tonuri înalte[NT0]

2.6 *Codoare cu predicție liniară cu analiză-prin-sinteză, folosind filtrul de ponderare perceptuală*

Putem acum să descriem primul codor de vorbire bazat pe cadre cu analiză-prin-sinteză, multipuls LPC[14]. Figura 9 este o schemă bloc simplificată a acestui codor. Diferența între semnalul de vorbire, s_n, și vorbirea de după cuantizare, $\hat{s}_n$, este calculată și trecută prin filtrul de ponderare perceptuală. Este calculată eroarea medie pătratică, MSE (*mean square error*), a semnalului diferență ponderat perceptual. Impulsurile sunt selectate în funcție de capacitatea lor de a minimiza eroarea medie pătratică.

Structura de mai sus poate fi modificată într-o formă matematică echivalentă prin calcularea semnalului diferență în spațiul perceptual. Filtrul LPC și filtrul de ponderare perceptuală pot fi combinate într-un singur filtru pentru semnalul de excitație. Ieșirea lui are două componente. Prima este răspunsul la intrare-zero. Aceasta este cauzată de memoria din LPC și de filtrul de ponderare perceptuală, când semnalul de intrare este adus la zero. I se mai spune și „sunetul filtrelor[16]" (*ringing out the filters*). Acesta, scăzut din semnalul de vorbire ponderat perceptual, va crea un vector țintă. Problema

---

[NT0] În cele ce urmează, determinarea sau predicția tonului are scopul de a determina dacă sunetele vorbirii, sau mai bine spus segmentele de vorbire care formează un cadru de eșantioane, se află în zonele spectrale dominante (unde zgomotele se maschează ușor) sau în văile spectrale (unde este necesară o protecție suplimentară la zgomot).

care apare atunci este de a găsi un set de impulsuri care, trecute fiind prin combinația dintre filtrul LPC și filtrul de ponderare perceptuală, plecând de la o stare inițială de zero total, să conducă la diferența minimă față de vectorul țintă. Găsirea setului de impulsuri poate fi ușurată prin corelarea vectorului țintă cu răspunsul net la impuls al filtrelor combinate. Sunt alese pozițiile impulsurilor care au corelația cea mai mare.

Un număr de inovații a urmat acestui model de bază. Întâi, a fost adăugat un predictor de tonuri (*pitch predictor*), ca în figura 10.[15] S-a descoperit că o așa zisă căutare de tonuri înalte cu buclă închisă are efectul cel mai bun. În această variantă, semnalul de excitație precedent este filtrat de filtrul LPC și de filtrul de ponderare și apoi corelat cu primul vector țintă. Segmentul care are cea mai mare corelație este selectat, scalat cu un vector de câștig (înmulțire scalară) și scăzut din primul vector țintă pentru a crea un al doilea vector țintă. Impulsurile sunt selectate astfel încât să minimizeze eroarea cu al doilea vector țintă. O limitare a predictorului de tonuri cu buclă închisă este aceea că este dificil să se caute o valoare a tonului, care este mai mică decât dimensiunea vectorului, așa cum se întâmplă adesea în cazul vocilor ascuțite. O îmbunătățire ulterioară a fost introducerea unui sistem de coduri adaptive numit „agendă de codare[16]" (*codebook*), sau matrice de codare. Întâi matricea de codare este încărcată cu semnalele anterioare de excitație. În eventualitatea că durata tonului este mai mică decât mărimea vectorului, matricea de codare scalează valorile precedente din cadrul vectorului. O metodă în plus de a îmbunătăți predictorul de tonuri este de a utiliza fracțiuni de semnal)[17] sau prelevări multiple pentru predictorul de ton.[18] Aceste metode sunt o consecință a faptului că spațierea uniformă în domeniul timp nu este echivalentă cu spațierea uniformă în domeniul frecvență. Ambele metode oferă însă o rezoluție bună în domeniul frecvență.

Două standarde se bazează pe codoarele LPC multipuls. Primul este „Airphone", folosit pe liniile aeriene comerciale. El folosește un codor multipuls de 9,6 kbps. Al doilea standard este Recomandarea G.723.1 a ITU. Ea conține un codor multipuls de 6,3 kbps.

La jumătatea anilor '80, când a fost publicat pentru prima dată codorul multipuls, complexitatea lui a fost considerată mare. O alternativă de complexitate mai redusă a fost creată prin codorul cu excitație de impulsuri regulate (RPE = *Regular Pulse Excitation*).[19] În codorul RPE impulsurile pot fi localizate numai în locurile prestabilite ale unui număr mic de grile. Care grilă trebuie folosită se determină după cum urmează. Semnalul de vorbire de intrare se filtrează de către inversul unui filtru LPC. Semnalul de ieșire rezultat este cunoscut ca semnalul rezidual LPC. (Dacă excitația ar fi semnalul rezidual, semnalul de vorbire de la ieșire ar fi exact copia originalului.) Dacă se folosește un predictor de tonuri, rezidualul LPC ar trebui, de asemenea, filtrat cu filtrul lui invers. Semnalul rezultat este apoi decimat cu un factor M, unde M este spațierea între punctele grilei, extrăgându-se un singur eșantion din M. Există M faze posibile pentru rezultatul decimării, în funcție de alegerea primului eșantion păstrat. Se selectează faza cu cea mai mare energie. Cuantizarea amplitudinii impulsurilor este făcută prin cuantizare scalară astfel încât să fie minimizată eroarea ponderată perceptual.

Codorul RPE a fost ales pentru prima generație de sistem de telefonie celulară european. El are M = 3 localizări diferite posibile pentru grile. În general rata este de 13 kbps și complexitatea este mai mică decât la standardele multipuls descrise mai sus.

Următoarea inovație s-a făcut în direcția unei complexități mai mari. Schroeder și Atal au publicat prima lucrare despre codarea cu predicție liniară cu excitație prin

matricea de codare[20] (CELP = *Codebook Excited Linear Prediction*). În esență, CELP este o codare multipuls folosind o cuantizare vectorială pentru localizarea și amplitudinea impulsurilor. Fiecare dintre vectorii din matricea de codare este filtrat de combinația dintre filtrele de sinteză și de ponderare și apoi corelat cu vectorul țintă. Din nou, vectorul din matricea de codare cu cea mai bună corelație este selectat și se alege o valoare de amplificare, care să minimizeze eroarea perceptuală.

În lucrarea originală despre CELP, matricea de codare consta din zgomote gaussiene albe preudo-aleatoare, cu variații de o unitate. Acest codor este cunoscut și sub denumirea de codor cu excitație stocastică, deoarece excitația părea să provină dintr-un proces stocastic de zgomot. Complexitatea acestui codor era (și este încă) extrem de mare, mai ales datorită detaliilor implicate de căutarea în matricea de codare. A apărut o întrecere în a găsi alte structuri de matrice de codare, care să reducă din complexitatea căutării.

Un lucru a fost repede descoprit și anume că se îmbunătățea calitatea dacă vectorii din matricea de codare erau „rari", adică aveau un conținut mai mare de zerouri și numai puține impulsuri. S-a descoperit că folosind zgomotul gaussian limitat central (*center clipped gaussian noise*) se obține un sunet mai bun. (Într-un fel, aceasta face ca semnalul de excitație să semene mai mult cu o secvență de impulsuri.) Faptul că matricea de codare era „rară", a putut fi utilizat și pentru a reduce calculele necesare căutării în matrice. Au fost introduse așa numitele matrice cu valori ternare, ale căror elemente sunt -1, 0 și 1, pentru a reduce calculele. Ordonarea (*overlapping*) matricei astfel încât al *n*-lea vector să fie compus în majoritate din segmente identice cu ale vectorului (*n-1*) și din numai puține elemente noi, a redus, de asemenea, timpul de calcul fără a diminua calitatea vorbirii la ieșire. Aceste trei tehnici, folosirea zgomotului gaussian limitat central, apoi cuantizarea elementelor ne-nule cu –1 și 1 și folosirea agendei compuse din aceste elemente (parțial ordonate), au fost combinate pentru a forma structura unei matrice de codare pentru un codor CELP pentru 4,8 kbps, utilizat pentru terminale de voce secretizată și cunoscute acum ca Standardul Federal al Statelor Unite FS-1016.[21]

Alții au încercat să micșoreze complexitatea prin utilizarea unor matrice de codare specializate, în ideea că va fi necesar să fie stocați și căutați mai puțini vectori. O altă metodă de a reduce complexiatea este de a folosi matrice de codare structurate. Codorul VSELP (*Vector Sum Excitation Linear Prediction*)[22] combină aceste două metode. Există numai câțiva vectori specializați, numiți vectori de bază (maxim 10) pentru o matrice de codare. Vectorii matricei de codare sunt combinații liniare ale acestor vectori de bază. Coeficienții acestor combinații liniare sunt –1, 0 și 1 și fiecare vector de bază este inclus în matrice. Vectorii de bază sunt filtrați de combinația dintre filtrul LPC și filtrul de ponderare. Apoi ieșirile sunt ortogonalizate între ele. (Dacă se folosește o matrice adaptivă, ieșirile trebuie ortogonalizate, de asemenea, cu vectorul selectat.) Apoi, acești vectori sunt corelați cu vectorul țintă. Căutarea vectorului compus se poate realiza foarte eficient dacă se folosește o secvență de coduri pseudo-Gray. Corelațiile individuale ale vectorilor de bază ortogonalizați pot fi numai adunate sau scăzute în evaluarea fiecărul vector de excitație posibil. VSELP a fost ales inițial ca standard Nord American de codare a vorbirii pentru telefonie celulară digitală, IS-54, ulterior ca standard japonez, iar mai târziu ca standard european de telefonie celulară digitală cu rată înjumătățită. Codorul IS-54 folosește două agende fixe de coduri. Această idee de a folosi etape multiple în căutarea semnalului de excitație constituie o altă cale de a reduce complexitatea.

Un alt exemplu de utilizare a matricei de codare structurate pentru a reduce complexitatea este CELP algebric (ACELP = *Algebraic CELP*)[23]. În această paradigmă, pozițiile posibile ale impulsurilor au fost împărțite în seturi disjuncte. Fiecare set poate contribui cu un singur impuls. Toate impulsurile au aceeași amplitudine, așa că nu trebuie transmise decât un bit de semn și un index al localizării impulsului. O singură valoare a amplificării este cuantizată și transmisă. „Rărimea" precum și structura agendei fac căutarea extrem de eficientă, în ciuda faptului că se pot utiliza un număr relativ mare de biți pentru a descrie impulsurile selectate. Se poate considera ACELP ca o versiune mai structurată a LPC multipuls. Structura matricii de codare ACELP este baza a două recomandări ITU. Una se referă la un codor de telefonie cu întârziere mică (mărimea cadrului de 10 ms) la 8 kbps (G.729). Cealaltă este opțiunea pentru rată mai mică, G.723, la 5,3 kbps.

În toate exemplele menționate mai sus, mărimea sub-cadrului era de 40 sau 60 eșantioane (5 sau 7,5 ms). Pentru un codor cu întârziere foarte mică, s-a constatat că agendele de coduri specializate se comportă cel mai bine. Codorul CELP[24] cu întârzierea redusă are o mărime a subcadrului de numai 5 eșantioane. El a devenit baza Recomandării G.728 a ITU. În timp ce pe ansamblu codorul este de complexitate mai mare decât alte codoare menționate, aceasta se datorează numai structurii adaptive din spate, care forțează filtrul LPC să fie actualizat la fiecare 2,5 ms și să se repete calculele la decodor la fel. Dar căutarea în matricea de codare nu este deloc mai complexă decât la alte codoare, datorită vectorilor foarte scurți.

2.7. *Alte utilizări ale mascării perceptuale*

După ce am explicat cum a ajutat filtrul de ponderare perceptuală la depășirea unor limite ale codării vorbirii, vom nota o a doua contribuție la îmbunătățirea calității codoarelor. Toate codoarele de vorbire date ca exemplu în secțiunile precedente produc încă distorsiuni perceptibile. Filtrul de ponderare perceptuală ajută la reducerea importantă a acestor distorsiuni. Distorsiunile sunt auzite mai mult în văile spectrale (subbenzi de frecvență cu componente spectrale rare și de mică amplitudine).

Ideea unui postfiltru adaptiv bazat pe filtrul de sinteză cu predicție liniară, folosit pentru a reduce zgomotul perceptibil, a fost introdusă în [25]. Postfiltrul adaptiv este un filtru liniar adițional care este combinat cu filtrul de sinteză pentru a obține rezultatul dorit. Scopul este de a reduce zgomotul în văile spectrale. Încă o dată, singura estimare a spectrului vorbirii avută la dispoziție la decodor este filtrul de sinteză. Ca și în cazul filtrului de ponderare perceptuală, filtrul de sinteză este modificat. În [26] această idee a fost extinsă pentru a include atât filtrele de durată scurtă (dominante), cât și pe cele de durată lungă (tonurile înalte). O problemă pe care au avut-o aceste metode de început a fost aceea că reducerea zgomotului se făcea pe seama calității vorbirii, devenită mai „înfundată". Aceasta se datora postfiltrelor care aveau o pantă spectrală ce favoriza frecvențele joase. În [27] filtrele de durată scurtă și cele de durată lungă au fost combinate cu un filtru de ordinul întâi, pentru a compensa panta de mai sus. Forma caracteristicii postfiltrului de durată scurtă a ajutat, de asemenea, la reducerea problemei spectrale date de pantă. O strategie de control al amplificării a ajutat la evitarea ca unul dintre segmente să fie prea tare sau prea slab. Postfiltre adaptive sunt folosite pentru fiecare dintre standardele menționate mai sus.

Ca o ilustrare a modului în care sunt construite filtrele adaptive, îl vom descrie pe cel utilizat în G.728. El constă dintr-un filtru LPC de ordinul 10, un postfiltru adaptiv

de tonuri înalte, un filtru spectral pentru „pantă" și un control al amplificării. Postfiltrul de durată scurtă, bazat pe filtrul LPC de ordinul 10, este caracterizat de:

$$H_s(z) = \frac{1 - A(z/\alpha_1)}{1 - A(z/\alpha_2)} \tag{9}$$

unde α_1 și α_2 sunt 0,65, respectiv 0,75. Funcția filtrului de înclinare spectrală este de a împiedica filtrul de durată scurtă să facă vorbirea prea înfundată. Aceasta se poate întâmpla datorită postfiltrului pe termen scurt, care tinde să aibă o pantă spectrală care favorizează frecvențele joase pentru vocale. Filtrul spectral pentru pantă este caracterizat de

$$H_L(z) = 1 + \alpha_3 k_1 z^{-1} \tag{10}$$

unde α_3 (=0,15) este un alt parametru fix de control, iat k_1 este primul coeficient de reflexie al filtrului LPC. Primul coeficient de reflexie este folosit pentru a contrabalansa panta spectrală dominantă, iar α_3 reglează nivelul contrabalansării. Postfiltrul de durată lungă este activat numai dacă există o corelație de cel putin 0,6 între vorbirea din cadrul curent și cea din cadrul de tonuri anterior. Se evită în acest fel adăugiri periodice la pauzele de vorbire (secvențe fără voce). Caracteristica acestui filtru este:

$$H_l(z) = g_l\left(1 + bz^{-P}\right) \tag{11}$$

unde P este perioada semnalului de ton înalt, iar b și g_l sunt bazați pe valoarea corelației. O schemă bloc a acestui postfiltru este ilustrată în figura 11. Semnalul de intrare în postfiltru, $s(n)$, este procesat întâi de postfiltrul de durată mare (pentru tonuri înalte), iar ieșirea acestuia este trecută prin postfiltrul de durată scurtă. Postfiltrul de durată scurtă constă dintr-un filtru având o parte pentru zona dominantă și o parte pentru compensarea pantei. Mărimea medie a semnalului $s(n)$ și cea de la ieșirea filtrului de compensare a pantei, $y(n)$, sunt comparate și se aplică o amplificare g_e lui $y(n)$, astfel încât mărimea medie a vorbirii postfiltrate, $s_{pf}(n)$, să fie aceeași cu a semnalului $s(n)$.

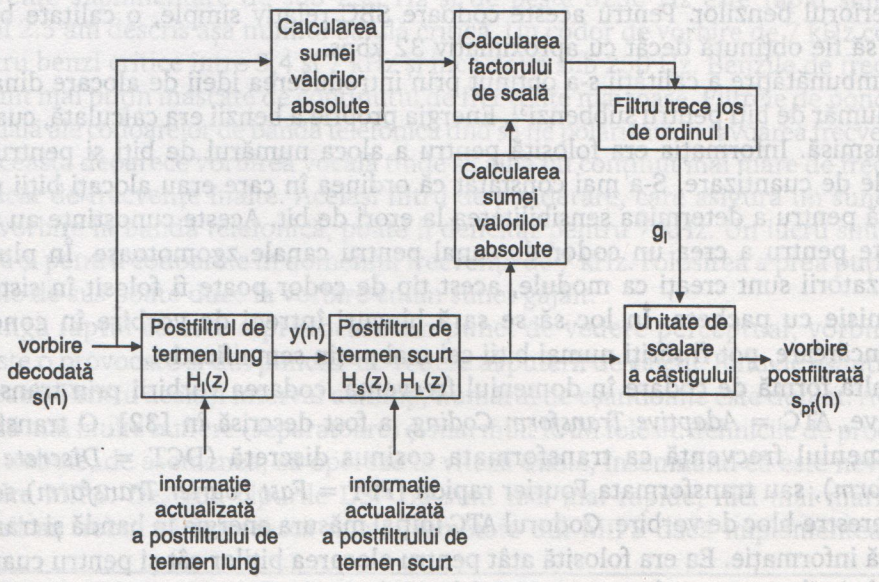


Figura 11. Schema bloc a postfiltrului adaptiv așa cum este implementată în Recomandarea G.728 a ITU

Aceasta ne completează imaginea codoarelor cu analiză-prin-sinteză şi a folosirii producerii vorbirii precum şi a percepţiei auditive în codarea vorbirii. Mai multe detalii despre acest domeniu al codării vorbirii pot fi găsite în lucrarea [28].

2.8. *Codoare de vorbire în domeniul frecvenţă*

În secţiunile precedente am discutat despre codoarele de vorbire cu predicţie liniară. Toate cuantizările semnalelor de excitaţie au fost realizate în domeniul timp. O altă clasă a codoarelor formelor de undă este cea a codoarelor de vorbire în domeniul frecvenţă. Modelul de bază este transformarea semnalului în domeniul frecvenţă şi cuantizarea în acest domeniu. La decodor, valorile din domeniul frecvenţă sunt refăcute şi transformate din domeniul frecvenţă în domeniul timp, pentru a sintetiza semnalul de vorbire la ieşire.

Unul dintre principalele avantaje ale codării în domeniul frecvenţă este acela că cea mai mare parte a înţelegerii noastre referitoare la mascarea perceptuală se bazează pe experimente realizate în domeniul frecvenţă. Folosirea unei reprezentări în domeniul frecvenţă a semnalului vorbirii, ne dă posibilitatea de a utiliza aceste cunoştinţe direct.

Cel mai simplu codor în domeniul frecvenţă este codorul de subbandă SBC (*Sub-Band Coder*). Semnalul de vorbire este descompus în câteva subbenzi de frecvenţă. În primele codoare SBC erau cam 5 benzi[29,30]. În fiecare bandă, semnalul din domeniul timp era prelevat selectiv (decimat) cu un factor potrivit. Reprezentarea în domeniul frecvenţă nu ar trebui să necesite o rată de eşantionare mai înaltă decât cea din domeniul timp. În aceste prime codoare, numărul de biţi utilizaţi pentru cuantizarea eşantioanelor era fix, bazat pe o mediere astfel încât să se obţină o calitate perceptuală. De cele mai multe ori, s-a folosit un cuantizator adaptiv-recursiv PCM. Predicţia adaptivă nu aduce un câştig mare pentru eşantioanele de bandă îngustă, deoarece într-o bandă îngustă sunetul este relativ alb. Datorită decimării era dificil să se obţină un predictor de tonuri în interiorul benzilor. Pentru aceste codoare SBC relativ simple, o calitate bună nu putea să fie obţinută decât cu aproximativ 32 kbps.

O îmbunătăţire a calităţii s-a obţinut prin introducerea ideii de alocare dinamică a unui număr de biţi pentru subbenzi[31]. Energia proprie a benzii era calculată, cuantizată şi transmisă. Informaţia era folosită pentru a aloca numărul de biţi şi pentru a seta nivelele de cuantizare. S-a mai constatat că ordinea în care erau alocaţi biţii putea fi folosită pentru a determina sensibilitatea la erori de bit. Aceste cunoştinţe au putut fi aplicate pentru a crea un codor de canal pentru canale zgomotoase. În plus, dacă cuantizatorii sunt creaţi ca module, acest tip de codor poate fi folosit în sisteme de transmisie cu pachete. În loc să se sară blocuri întregi de vorbire în condiţii de supraîncărcare, pot fi săriţi numai biţii cei mai puţin semnificativi.

O altă formă de codare în domeniul frecvenţă, codarea vorbirii prin transformări adaptive, ATC = *Adaptive Transform Coding*, a fost descrisă în [32]. O transformare în domeniul frecvenţă ca transformata cosinus discretă (DCT = *Discrete Cosine Transform*), sau transformata Fourier rapidă (FFT = *Fast Fourier Transform*) se aplică unei ferestre-bloc de vorbire. Codorul ATC iniţial măsura energia în bandă şi transmitea această informaţie. Ea era folosită atât pentru alocarea biţilor cât şi pentru cuantizare, în mod similar cu metoda pentru SBC folosită în [31]. (De notat că folosirea în ATC a precedat-o pe cea din SBC).

O a doua investigare a ATC folosea informații suplimentare bazate pe o analiză LPC a semnalului de vorbire de intrare[33,34]. Folosind o transformare a coeficienților LPC, spectrul de frecvențe putea fi prezis. S-a introdus un predictor de tonuri înalte, astfel încât structurile fine de frecvențe ale vorbirii sonore să poată fi, de asemenea, prezise. Folosind acest model de producere a informației, rata de bit necesară unei transmisii de bună calitate a fost redusă. O a treia investigație asupra ATC folosește un model cepstral[NT1] al producerii vorbirii, pentru a implementa ATC în timp real într-o structură de procesor[35].

Au mai fost și alte studii atât asupra SBC cât și asupra ATC.[36-39] Oricum, fie calitatea tuturor acestor codoare în domeniul frecvență a fost mereu mai mică decât a corespondentelor lor în domeniul timp din aceeași perioadă, fie complexitatea sau întârzierea au fost mai mari. Recent, aceste codoare au fost reconsiderate printr-o nouă întorsătură. A fost aplicată paradigma (modelul) analizei-prin-sinteză. Filtrul LPC uzual și predictorul de tonuri formează etajul inițial al codorului. După calcularea reziduului, el este transformat și cuantizat în domeniul frecvență. Aceasta stă la baza codorului TCX (*Transform Coded Excited*)[40,41].

O altă orientare în codarea în domeniul frecvență este introducerea unei analize a transformărilor undelor mărginite (*wavelet* [NT2]). Blocurile de filtrare a undelor mărginite (în timp și frecvență) au banda de filtrare cea mai plată. Tehnicile de cuantizare folosite de analiza undelor mărginite (în timp și frecvență) sunt destul de asemănătoare cu cele aplicate înainte pentru SBC și ATC. Îmbunătățirea calității se datorează unor filtre mai bune (cu analiză-prin-sinteză).

2.9. *Codarea vorbirii de bandă largă (7 kHz)*

Codarea vorbirii de bandă largă (7 kHz) prezintă câteva diferențe importante în comparație cu vorbirea limitată la banda telefonică. Este foarte ușor pentru orice ascultător să distingă vorbirea de bandă largă de cea de bandă telefonică. Conținutul de frecvențe suplimentare de sub 250 Hz și de peste 3250 Hz este rapid simțit. În capitolul 2.5 am descris așa numita bandă critică. Un codor de vorbire de 7 kHz conține încă patru benzi critice între 3,4 și 7 kHz și încă două sub 250 Hz. Benzile de frecvențe înalte sunt mai puțin mascate de conținutul de frecvențe mai joase. Filtrele de ponderare perceptuală ale codoarelor de bandă telefonică tind să fie polarizate în favoarea frecvențelor joase. Aceasta deoarece vorbirea vocală tinde să aibă un conținut mai mare de frecvențe joase decât de frecvențe înalte. Același filtru de ponderare, care asigură un sunet mai bun de vorbire în banda telefonică, poate fi deficitar pentru 7 kHz. Un lucru similar se întâmplă și pentru codoarele în domeniul frecvență de 7 kHz. Folosirea a prea puțini biți în benzile de sus poate duce la vorbire cu un sunet gâjâit.

Pe lângă faptul că este o provocare din punct de vedere perceptual, vorbirea de 7 kHz este o provocare și din punctul de vedere al puterii de calcul. Rata de eșantionare se dublează. Pentru același interval de timp, numărul de eșantioane este dublat. Aceasta înseamnă mai multe buffere (separatoare) și mai mult RAM folosit. Tehnicile de procesare folosite trebuie, de asemenea, să opereze la viteză dublă, însemnând că este nevoie de mai multe MIPS. Desigur, cipurile DSP nu sunt nici mai rapide, nici mai mari, deci algoritmii ar trebui să folosească numai jumătate din MIPS dacă implementează un

---

[NT1] Cepstrum este rezultatul unei inversări spectrale, de unde și denumirea (spec-trum, ceps-trum).

[NT2] „Wavelet" este o formă de undă mărginită atât în timp cât și în frecvență.

codor pentru banda telefonică. Acest lucru este şi mai dificil deoarece pentru a modela spectrul vorbirii este necesar un predictor liniar de ordin mai mare. În locul analizei LPC de ordin 10, care se foloseşte pentru vorbirea în banda telefonică, utilizăm adesea analiza de ordinul 16.

Vorbirea de 7 kHz a fost codată folosind ambele tehnici, în domeniul timp şi în domeniul frecvenţă. Singurul standard actual pentru codarea vorbirii de 7 kHz este Recomandarea G.722 a ITU, care este descrisă în capitolul 3.1. El foloseşte un filtru oglindă în cuadratură cu 24 de prize, pentru a diviza semnalul în două benzi. Fiecare este codată ADPCM şi transmisă. Filtrul cu 24 de prize creează o întârziere de numai 1,5 ms şi este implementat eficient folosind numai 12 înmulţiri pe eşantion prelevat (în urma decimării). Cu numai două benzi, acest codor este numai cu greu considerat un codor de vorbire în domeniul frecvenţă. Înalta fidelitate a codării benzii de jos (până la 6 biţi pe eşantion) maschează foarte bine codarea aproape „scârţâită" pe 2 biţi a eşantioanelor din banda de sus. Acest exemplu arată că sunt posibile unele mascări perceptuale între banda telefonică şi frecvenţele mai mari. În acelaşi timp, el pare să indice că răspunsul urechii este favorabil la conţinutul de frecvenţe înalte, chiar dacă acestea sunt puţin scârţâite.

Sunt câteva exemple notabile de codare în domeniul timp al vorbirii de 7 kHz. Toate folosesc codarea vorbirii cu analiză-prin-sinteză. Excitaţia ACELP (*Algebraic CELP*) a fost folosită în [42] şi [43]. Codorul de vorbire al Norwegian Telecom este un codor CELP de bandă largă.[44] Acest codor a fost testat oficial şi calitatea lui la 16 kbps este între cele ale codoarelor G.722 de 56 şi 64 kbps. Ambele decodoare folosesc LPC adaptive-direct (nerecursive) şi informaţia de tonuri cu dimensiunea cadrelor de 20 ms şi mai mult.

Un codor CELP de vorbire de bandă largă cu întârzirere mică este descris în [45] şi [46]. S-a folosit un filtru LPC de ordinul 32. Coeficienţii lui sunt derivaţi adaptiv-recursiv. El folosea o dimensiune a cadrului de 5 eşantioane, dând o întârziere la comunicaţia într-o direcţie de sub 1 ms. Aceasta este o întârziere mai mică decât a codorului G.722. Oricum, pentru a obţine o calitate înaltă a vorbirii, este necesară o rată de bit de 32 kbps.

Sunt două probleme la utilizarea în domeniul timp a versiunilor modificate ale codoarelor de 3 kHz pentru a coda vorbirea de bandă largă: ori vorbirea sună înfundat, ori frecvenţele înalte sunt zgomotoase. În aceste trei codoare, au fost folosite două abordări: modificarea filtrului de ponderare perceptuală şi preaccentuarea frecvenţelor înalte înaintea codării în vederea îmbunătăţirii cuantizării lor. În ultimul caz, o dezaccentuare echivalentă s-a aplicat la decodare, înainte de ieşire. Rezultă o vorbire mai zgomotoasă la frecvenţele înalte.

Codorul TCX menţionat mai sus este un codor de vorbire de bandă largă. Cum s-a indicat mai înainte, el reprezintă o încercare de a combina codarea transformatei pentru semnalul de excitaţie şi modelul LPC (codare cu predicţie liniară). În acest moment, codorul este încă în dezvoltare şi ultimele sale performanţe nu sunt de fapt cunoscute.

2.10. *Codoare de vorbire bazate pe modele cu rată de bit scăzută*

Pe măsură ce încercăm să codăm vorbirea cu rate de bit din ce în ce mai reduse, ne bazăm din ce în ce mai mult pe un model al vorbirii. O rată de bit de 2,4 kbps corespunde la numai 0,3 biţi pe eşantion la o frecvenţă de eşantionare de 8 kHz. Este evident că nu

sunt destui biți disponibili pentru a încerca să reproducem forma de undă inițială sau măcar o aproximație a ei. Standardul cu rata cea mai joasă care se bazează încă pe o potrivire aproximativă a formelor de undă este PSI-CELP de 3,45 kbps (vezi capitolul 3.4 pentru mai multe detalii).

Rata redusă implică, de asemenea, întârzieri în încercările noastre de a coda vorbirea. Pentru a transmite parametrii unui model trebuie să avem un număr suficient de biți. De exemplu, multe cuantizatoare LPC reprezintă informația pentru 20-30 de biți pe cadru. Dacă alocăm 1200 bps pentru a reprezenta informația spectrală și folosim un cuantizator LPC pe 30 de biți, putem transmite informații de 40 de ori pe secundă. Aceasta înseamnă că dimensiunea cadrului ar fi de 25 ms. Aceasta sugerează că este foarte improbabil ca un codor de vorbire cu rată de bit redusă și cu o întârziere mică să poată oferi o bună calitate a vorbirii. (Codorul PSI-CELP menționat mai sus are o durată a cadrului de 40 ms.)

Acest capitol este dedicat codărilor cu tehnici bazate pe modele. Vocoderul[1] original a fost un codor de vorbire bazat pe modele. Vocoderul LPC descris în capitolul 2.3 a fost un alt codor de vorbire bazat pe model. În ambele codoare, nu exista nici o încercare de a reproduce o replică a semnalului de intrare. În schimb, ele încercau să producă o formă de undă cu proprietăți statistice similare cu ale semnalului de vorbire de la intrare. Modelul de codare cu predicție liniară a fost discutat pe larg în capitolele anterioare despre codoarele formelor de undă. Acest capitol se va concentra pe alte paradigme de codare bazată pe modele.

În ultimii ani au evoluat două metode de codare în domeniul frecvență, bazate pe modele. Ele sunt codarea cu transformare sinusoidală (STC = *Sinusoidal Transform Coding*)[47,48] și codarea cu excitație multi-bandă (MBE = *Multi-Band Excitation*)[49,50]. Cele două tehnici sunt strâns legate. Ambele codoare folosesc sinteza sinusoidală pentru toate armonicile. Ambele tehnici realizează o estimare a spectrului vorbirii și o transmit ca pe o informație colaterală. Codorul STC a luat în considerare folosirea unei modelări LPC de ordin superior, precum și alte metode parametrice de cuantizare a acestei informații. Cele mai multe versiuni de codor MBE nu folosesc LPC pentru informația suplimentară. Codorul STC pleacă de la premisa că semnalul de vorbire poate fi produs ca o sumă de sinusoide legate armonic. Dacă luăm în considerare spectrul vorbirii fără pauze, atunci fazele acestor armonici sunt continue de la un cadru la următorul. Oricum, dacă vorbirea nu conține voce, atunci vorbirea poate fi încă modelată ca o sumă de sinusoide, dar faza este discontinuă. Inițial, codorul STC se baza pe decizia asupra prezenței sau absenței vocii. O îmbunătățire ulterioară a fost considerarea faptului că spectrul este împărțit în părți cu sau fără voce. În fiecare cadru se găsește o frecvență care divide spectrul. Toate armonicile de sub această frecvență sunt considerate ca aparținând vocii, în timp ce toate de deasupra ei sunt considerate frecvențe non-vocale.

Codoarele cu excitație multi-bandă duc acest model un pas mai departe. In extremis, fiecare frecvență armonică ar putea fi declarată ca aparținând vocii sau nu. În practică, numărul de biți necesar transmiterii acestei informații ar fi prea mare pentru un codor cu o rată de bit redusă. În schimb, armonicile sunt grupate în benzi și un singur bit reprezintă, pentru o bandă, decizia dacă ea aparține sau nu vocii. Ambele codoare folosesc un amestec de componente de excitație aparținând sau nu vocii, pentru a produce un sunet de vorbire mai natural decât dacă excitația ar fi fost considerată toată

aparținând vocii, sau toată ne-aparținând vocii. În practică aceasta constituie o mare îmbunătățire a calității în comparație cu decizia tradițională „voce / fără voce", pentru că ambele componente sunt prezente în majoritatea sunetelor vorbirii.

Ambele codoare (ca și altele bazate, de asemenea, pe modele) sunt vital dependente de o estimare corectă a frecvențelor tonurilor. Pe parcursul câtorva cadre, ele trebuie să urmărească schimbările naturale în frecvențele tonurilor. Calitatea percepută este direct legată de cât de bine este modelat tonul. Principala dificultate pe care o întâlnesc codoarele bazate pe modele apare atunci când vorbirea de intrare mai conține în fundal și alte semnale. O cantitate mare de zgomot de fundal sau muzică poate să afecteze determinarea cu acuratețe a tonurilor. În prezența unui fundal foarte zgomotos, determinarea tonurilor poate deveni imposibilă. Fundalul foarte zgomotos va influența, de asemenea, decizia dacă armonicile aparțin sau nu vocii. Codorul se poate bloca pe o decizie permanentă de tipul „fără voce". Aceasta are un impact serios asupra calității și a inteligibilității. Când semnalul de intrare este numai zgomot de fundal, modelul încă îl analizează, ca și când ar fi vorbire. Aceasta poate produce o componentă de voce în semnalul de ieșire, acolo unde nu era prezentă în semnalul de intrare. De fapt, codoarele bazate pe modele pot adesea să producă artefacte (produse false) sonore nenaturale, când semnalul de intrare nu este vorbire curată.

Un alt codor bazat pe modele, care a fost recent introdus, face uz de sinteza LPC. Este numit „cu interpolare a formelor de undă".[51,52] Formele de undă, care sunt modelate, sunt segmente ale perioadelor individuale ale tonurilor din semnalul de excitație. Nu există o decizie „cu voce / fără voce". În schimb, semnalul de excitație este descompus în două componente, una evoluând lent și care poate fi modelată ușor de structura de tonuri, cealaltă evoluând rapid și care poate fi modelată prin zgomot colorat. Deoarece componenta cu evoluție lentă are variații relativ mici de la un cadru la altul, modificările ei pot fi cuantizate diferențial cu relativ puțini biți. Deoarece urechea este insensibilă la faza zgomotului, componenta cu evoluție rapidă poate fi cuantizată cu numai câțiva biți pentru a indica forma spectrală aproximativă a zgomotului. Să ne amintim că suma dintre cele două semnale are un spectru relativ plat deoarece ele formează semnalul rezidual albit, obținut după filtrarea inversă LPC. În consecință, parametrii ambelor componente pot fi eficient cuantizați și transmiși.

Folosind modelul interpolării formelor de undă, zgomotul de fond este modelat ca făcând parte din componenta cu variație rapidă (REW = *Rapidly Evolving Waveform*). Aceasta cauzează o degradare a calității deoarece REW este utilizat pentru a modela atât porțiunile fără voce cât și zgomotul de fond al vorbirii. Pe măsură ce zgomotul devine mai puternic, apare și un impact asupra estimării tonurilor, făcând codorul să sune mai mult ca un vocoder, deoarece tonurile nu au fost suficient modelate.

Calitatea relativă a acestor codoare atât de recente, bazate pe modele, la o rată de bit între 2,4 și 4 kbps, este surprinzător de mare. Pentru un semnal curat de intrare, un codor îmbunătățit MBE operând la cca. 4 kbps a fost declarat ca având calitatea echivalentă cu prima generație de standarde celulare digitale, ca IS-54 VSELP[53], iar un codor de interpolare a formelor de undă operând la 2,4 kbps asigură în toate condițiile o calitate cel puțin echivalentă cu a codorului curent de voce cifrat, de 4,8 kbps.[54] Codorul MBE este deja baza standardului Inmarsat M și o nouă versiune a fost selectată ca standard Inmarsat Mini-M. În viitor, se așteaptă noi îmbunătățiri ale calității sunetului pentru codoarele bazate pe modele.

3. Standarde curente

Această parte este divizată în subcapitole despre standardele și activitățile de codare curente. Subcapitolele conțin informații despre codoarele care au fost sau vor fi în curând standardizate. Vom începe prin a descrie pe scurt organizațiile de standardizare care formulează standarde de codare a vorbirii și procedurile pe care le urmează pentru a realiza aceste standarde.

ITU (*International Telecommunications Union*) este o agenție a UNESCO (*United Nations Economic, Scientific and Cultural Organization*), însărcinată cu toate aspectele standardizării în rețele de telecomunicații și radio. Sediul său este la Geneva, în Elveția. La început, a fost împărțită în două organizații, CCITT (Comitetul Consultativ Internațional pentru Telegrafie și Telefonie[NT3]) și CCIR (Comitetul Consultativ Internațional pentru Radio[NT3]). În 1993 aceste organisme au fost reorganizate. ITU-T (*ITU Telecommunications Standardization Sector*) și-a asumat toate responsabilitățile pe care le-a avut inițial CCITT și câteva dintre cele asociate înainte cu CCIR. ITU-R (*ITU Radio Standardization Sector*) și-a asumat responsabilitățile care au mai rămas de la CCIR. ITU-T formulează standarde atât pentru comunicațiile cablate cât și pentru cele fără fir. ITU-R se ocupă de standardele legate de aspectele radio. Există și o a treia ramură, ITU-B (*ITU Telecommunications Standards Bureau*), care face toată administrarea birocratică a hârtiilor.

Standardele de codare a vorbirii sunt administrate împreună de Grupurile de Studiu 12 și 16 din cadrul ITU-T. Alte Grupuri de Studiu pot iniția cereri pentru codoare de vorbire pentru aplicații specifice. Experții în codarea vorbirii se găsesc în SG16 (SG = *Study Group*). Experții în calitatea vorbirii se află în SG12, într-un organism special numit SQEG (*Speech Quality Experts Group*). Când se lucrează la un nou standard, SG16 definește o listă de cerințe bazate pe aplicațiile dorite. SQEG și alte organisme interesate pot revizui cerințele înainte de a fi finalizate. SQEG creează apoi un plan de testări și o listă a laboratoarelor de testări subiective, care vor ajuta la măsurarea calității codoarelor de vorbire în diferite condiții. Procesul de standardizare consumă mult timp și poate dura între doi și șase ani.

Trei organizații diferite fac standarde regionale pentru telefonie celulară, inclusiv pentru codarea vorbirii. În Europa, forul superior este ETSI (*European Telecommunications Standards Institute*). ETSI este o organizație compusă în principal din producători de echipamente de telecomunicație. Ea a fost desprinsă din CEPT (*Conference of European Posts and Telephones*). Primul standard de telefonie digitală celulară a fost de fapt creat sub autoritatea CEPT, înainte de înființarea ETSI. Organismul însărcinat în mod curent cu elaborarea standardelor de codare digitală celulară a vorbirii este cunoscută ca TCH-HS. În America de Nord, organismul tutelar este ANSI – *American National Standards Institute*. Organismul însărcinat cu elaborarea standardelor pentru celulare digitale este TIA – *Telecommunications Industry Association*. În Japonia, organismul însărcinat cu elaborarea standardelor pentru celulare digitale este RCR – *Research and Development Center for Radio Systems*.

Mai există, de asemenea, standarde pentru codarea vorbirii pentru satelit, urgențe și telefonie cifrată. Unele dintre aceste standarde au fost promulgate de organisme guvernamentale, în timp ce altele au fost promulgate de organizații private.

[NT3] Denumirea originală este în limba franceză, de unde și potrivirea acronimului cu limba română.

Fiecare dintre aceste organizații de standardizare lucrează în conformitate cu propriile reguli și regulamente. Există însă o caracteristică comună a tuturor acestor organizații. Aceasta constă în procesul de elaborare a standardelor. Întâi trebuie să se obțină un consens asupra necesității standardului. În cele mai multe cazuri aceasta este o evidență. De exemplu, în telefonia celulară digitală, este necesar un standard foarte cuprinzător, astfel încât diferitele componente ale sistemului să poată lucra împreună. Puține companii din lume ar putea fi în situația de a fabrica echipamente pentru toate părțile sistemului. Elaborarea unui standard este un proces lung, care nu trebuie tratat cu superficialitate.

În al doilea rând, trebuie creați termenii de referință. Acesta devine documentul care guvernează întregul efort. Întâi, el definește aplicațiile care sunt avute în vedere. (De fapt, această secțiune oferă motivele pentru care este necesar standardul.) Pe baza aplicațiilor, se pot formula cerințe pentru atributele codorului de vorbire: calitatea, complexitatea, rata de bit și întârzierea. Cerințele vor determina mai târziu programul de testări care va trebui să determine dacă diverși candidați sunt corespunzători. În sfârșit, membrii grupului trebuie să-și propună un calendar de urmărit.

Trebuie să existe o perioadă inițială în care să se permită promotorilor să proiecteze codoarele care se vor supune cerințelor. În ITU, promotorilor li se cere să se supună unor teste de calificare, care să determine în ce măsură codorul lor candidat ar putea răspunde cerințelor. Se stabilește un termen limită pentru prezentarea la teste. Trebuie definit exact ceea ce promotorii trebuie să obțină prin codoarele lor candidat. În același timp, trebuie asigurate serviciile unuia sau mai multor laboratoare de testare și trebuie definit un plan de testări. Un laborator gazdă trebuie, de asemenea, să proceseze toate datele care vor fi folosite în testele subiective.

O dată terminate testele subiective, trebuie stabilite criteriile după care se face selecția. Uneori se folosesc formule care includ atât rezultatele testelor subiective de calitate cât și unele măsurători ale complexității. În ITU, sunt de obicei mai puțini candidați și se folosesc baremuri. Un codor candidat trece de barem sau nu.

Odată selecția făcută, trebuie scris un proiect de standard. În unele cazuri, ca și la ITU, standardul este foarte specific, toate aspectele codorului sunt clar definite și trebuie implementate pentru a se conforma. Avantajul acestei abordări este că pentru toate implementările codorului de vorbire, calitatea va fi aceeași. În alte cazuri, cum este celularul digital IS-54, se folosește numai un standard de flux al biților. Algoritmul este definit, dar multe detalii sunt lăsate la alegerea producătorului. Numai fluxul de biți este definit. Calitatea diferitelor implementări va fi diferită și nu va exista o limită minimă garantată pentru calitate, dar nu va exista, totodată, nici un impediment pentru a îmbunătăți calitatea. Alegerea uneia sau alteia dintre abordări va determina și cât de extinse vor fi instrumentele de verificare a conformității. Este extrem de important să se realizeze unele instrumente de verificare, astfel încât agențiile de service și clienții să poată ști dacă un anumit codor se conformează unui standard și va funcționa.

În subcapitolele următoare vom trece în revistă cele mai importante standarde de codare a vorbirii ale diferitelor organizații de standardizare. Începem cu ITU-T, împărțind lucrările lor în trei subcapitole. Vom revedea standardele pentru telefonie celulară și în final câteva alte standarde. La sfârșit vom oferi câteva informații despre câteva standarde de codare a vorbirii, produse de alte organizații. În fiecare caz, vom arunca o scurtă privire asupra atributelor codoarelor și a tehnicii pe care se bazează. O descriere amănunțită a acestor codoare poate fi găsită în referințele bibliografice.

3.1. *Codoarele curente ITU pentru semnale cu formă de undă*[NT4]

Tabelul 3.1 descrie recomandările ITU pentru codarea vorbirii, care se bazează pe cuantizarea scalară eşantion cu eşantion. Trei dintre aceste codoare operează în domeniul timp pe semnalul original eşantionat, în timp ce al patrulea se bazează pe codarea cu două subbenzi pentru vorbirea de bandă largă.

Tabelul 3.1. Codoare ITU pentru vorbire

Organizaţia de standardizare	ITU	ITU	ITU	ITU
Recomandarea	G.711	G.726	G.727	G.722
Tipul de codor	PCM compandat	ADPCM	ADPCM	SBC/ADPCM
Data	1972	1990	1990	1988
Rata de bit	64 kbps	16-40 kbps	16-40 kbps	48-64 kbps
Calitatea	Telefonie	≤ Telefonie	≤ Telefonie	Comentariu
Complexitatea				
MIPS	<< 1	1	1	10
RAM	1 octet	< 50 octeţi	< 50 octeţi	1K
Întârzierea				
Mărimea cadrului	0,125 ms	0,125 ms	0,125 ms	1,5 ms
Tipul specificaţiei				
Virgula fixă	Bit fix	Bit fix	Bit fix	Bit fix

CCITT a standardizat două codoare PCM compandate de 64 kbps în 1972. America de Nord şi Japonia folosesc PCM – legea μ. Restul lumii foloseşte PCM – legea A. Ambele codoare folosesc 8 biţi pentru a reprezenta semnalul. Raportul efectiv Semnal / Zgomot este de cca. 35 dB. Tabelele pentru ambele caracteristici de cuantizatoare G.711 sunt conţinute în [55]. Un codor PCM – legea μ are o gamă echivalentă cu un PCM liniar de 13 biţi, dar are o rezoluţie mai grosieră decât un PCM – legea A pentru valori mici ale eşantioanelor. Un codor PCM – legea A are o rezoluţie mai fină la eşantioane mici, dar gama lui este echivalentă cu a unui PCM liniar de numai 12 biţi. Ambele codoare sunt considerate echivalente din punct de vedere al calităţii pe ansamblu. O codare în tandem cu oricare dintre codoare este considerată echivalentă cu eliminarea celui mai puţin semnificativ bit (ceea ce ar echivala cu a reduce rata de bit la 56 kbps). Ambele codoare sunt extrem de sensibile la erori de bit în zona biţilor celor mai semnificativi. Complexitatea lor este foarte redusă.

Codorul de 32 kbps ADPCM G.721 a fost standardizat de ITU întâi în 1984.[7,56] Prima lui aplicaţie s-a dorit să fie un echipament de multiplicare cu circuit digital (DCME = *Digital Circuit Multiplication Equipment*). În combinaţie cu o interpolare pentru vorbire digitală, s-a obţinut o creştere de 5:1 la capacitatea cablurilor submarine şi a legăturilor prin satelit pentru conversaţii vocale. Un motiv în plus pentru crearea lui a fost acela că se întâmpla adesea ca astfel de legături să aibă la un capăt un PCM legea μ (şi la celălalt capăt un PCM – legea A. G.726 poate accepta atât un PCM – μ cât şi un PCM – A ca intrare sau ieşire. Oricum, nu a fost creat pentru a accepta un PCM liniar ca intrare.

[NT4] Prin exprimarea „semnale cu formă de undă" se exclud semnalele binare, de impulsuri, de natură fundamental diferită de cele specifice vorbirii.

Poate cea mai singulară trăsătură a lui este o proprietate numită „tandemizare sincronă" (*synchronous tandeming*).[57] Dacă un circuit implică două codări ADPCM având între ele un codor PCM – legea μ sau PCM – legea A, nu apare nici o degradare adiţională datorată celei de a doua codări. Al doilea flux de biţi va fi identic cu primul! G.721 a fost standardizat întâi în 1984 şi a trebuit să fie apoi re-standardizat în 1986 datorită unor probleme minore din prima versiune. O problemă a fost aceea că reţelele din America de Nord nu pot avea fluxuri lungi de 0-uri. Cuvintele de cod care conţin numai zerouri au fost eliminate din a doua versiune a G.721 prin schimbarea cuantizării de la 16 niveluri la 15. Cealaltă problemă implica trecerea unor anume semnale de modem. În forma lui finală, pot să treacă numai semnalele de modem de până la 4800 bps.

O recomandare de ADPCM cu numărul G.723 a fost aprobată în 1988. Ea a fost o extensie a G.721 pentru două rate adiţionale, 24 şi 40 kbps. 24 kbps este utilizată numai în DCME pentru condiţiile în care traficul este pe moment mai mare decât capacitatea transmisiei. 40 kbps este capabilă să accepte semnalele modemurilor de 9600 şi 14400 bps, care nu puteau fi codate satisfăcător de G.721.

Actuala recomandare G.726 este o unificare a G.721 şi G.723. O rată de 16 kbps este, de asemenea, inclusă pentru DCME în condiţiile de supraîncărcare. Codoarele de 16 şi 24 kbps nu sunt de calitate telefonică. Oricum, dacă ele nu sunt utilizate decât sporadic, pe durate scurte de supraîncărcare, calitatea în ansamblu a codorului de vorbire DCME este considerată totuşi de calitate telefonică. (O dată cu aprobarea G.726, recomandările anterioare G.721 şi G.723 au fost oficial abrogate).

G.727 include aceleaşi rate ca şi G.726, dar toţi cuantizatorii au un număr par de nivele. Cuantizatorul de 2 biţi este imbricat în cuantizatorul de 3 biţi, care este imbricat în cuantizatorul de 4 biţi, care este imbricat în cuantizatorul de 5 biţi. Acest lucru este necesar pentru PCME (*Packet Circuit Multiplex Equipment*) unde biţii cei mai puţin semnificativi din pachet pot fi abandonaţi în condiţii de supraîncărcare.

Recomandarea G.722 este un standard de codare de vorbire de bandă largă. Principalele lui aplicaţii sunt teleconferinţele şi videoteleconferinţele.[58] Banda lărgită (50-7000 Hz) are un sunet mai natural şi mai puţin obositor decât banda telefonică (200-3200 Hz). Banda mai mare măreşte inteligibilitatea vorbirii, mai ales pentru sunetele fricative „f " şi „s", care sunt dificil de distins pentru banda telefonică. Codorul G.722 este un codor cu două-benzi (subbenzi), cu codare ADPCM în ambele benzi. Structura ADPCM este similară cu cea a recomandării G.727. Banda de sus foloseşte un codor ADPCM cu un cuantizator adaptiv de 2 biţi. Banda de jos foloseşte un codor cu cuantizator adaptiv de 4-5-6 biţi imbricat. Aceasta face posibile oricare din ratele de 48, 56 şi 64 kbps. Un filtru oglindă în cuadratură cu 24 de prize este folosit pentru a împărţi eficient semnalul. În testele subiective de comparare cu vorbirea de bandă telefonică, banda de 7 kHz este totdeauna preferată, uneori chiar cu 1 MOS.

3.2. *Codoare de vorbire ITU cu predicţie liniară cu analiză-prin-sinteză*

Tabelul 3.2 descrie trei recomadări ITU de codoare cu analiză-prin-sinteză. Toate trei sunt codoare de blocuri, bazate pe extensia codorului de vorbire multipuls LPC.[14]

Tabelul 3.2. Codoare de vorbire ITU cu predicție liniară cu analiză-prin-sinteză

Organizația de standardizare	ITU	ITU	ITU
Recomandarea	G.728	G.729	G.723.1
Tipul de codor	LD-CELP	CS-ACELP	MPC-MLQ și ACELP
Data	1992, 1994	1995	1995
Rata de bit	16 kbps	8 kbps	6,3 & 5,3 kbps
Calitatea	Telefonie	Telefonie	≤ Telefonie
Complexitatea			
MIPS	30	≤ 20	≤ 15
RAM	2 K	< 2,5 K	2,2 K
Întârzierea:			
Mărimea cadrului	0,625 ms	10 ms	30 ms
Previziune	0	5 ms	7,5 ms
Tipul specificației			
Virgulă mobilă	Algoritm exact	Nu	Nu
Virgula fixă	Bit fix	Bit fix C	Bit fix C

ITU a definitivat în 1988 cerințele pentru o calitate telefonică a codorului de vorbire de 16 kbps. Prin calitate telefonică s-a înțeles calitatea similară sau superioară performanței lui G.726 ADPCM de 32 kbps. Intenția inițială a fost să se poată utiliza acest tip de codor oriunde în rețea. De aici cerințele pentru întârziere mică. Oricum, utilizarea lui universală a fost ulterior restrânsă deoarece nu putea să permită trecerea semnalelor de modem cu rate mai mari de 2400 bps. Codorul LD-CELP (*Low-Delay CELP* = CELP cu întârziere mică) a fost singurul candidiat care a demonstrat o calitate suficientă a vorbirii pentru a corespunde cerințelor unei calități riguroase de vorbire.[59] G.728 a fost inițial specificat ca un codor CELP cu virgulă mobilă,[23] care cerea producătorilor să urmeze cu exactitate algoritmul specificat în recomandare. S-a creat un set de vectori de test pentru a verifica implementarea corectă.[60] Ca urmare, o specificație pentru un algoritm cu virgulă fixă a devenit necesară și a fost terminată în 1994. Performanțele lui G.728 au fost testate extensiv de SQEG.[61] El asigură performanțe superioare pentru semnalele cu fundal zgomotos sau cu muzică. Este foarte rezistent la erori de bit aleatoare, mai mult decât predecesoarele standarde ITU G.711, G.726 și G.727 și decât standardele mai noi, descrise mai jos. În plus față de posibilitatea de a lăsa să treacă semnalele de modem cu rate de bit joase, de până la 2400 bps, lasă să treacă toate tonurile de semnalizare ale rețelei.

Ca o ironie, prima aplicație a lui G.728 a fost un videofon, H.320, cu rată de bit joasă, la care întârzierea nu era importantă. Este în curs, la SG16, specificarea unei extensii a lui G.728, care să poată fi aplicată în echipamente de multiplicare și în aplicații radio pentru canale cu fading care poate determina dispariția cadrelor.

Ca răspuns la o solicitare din partea Grupului de Lucru 8/1 al CCIR pentru un codor de rețele necablate destinat FPLMTS (*Future Public Land Mobile Telecommunication Service*), ITU a inițiat în 1990 un program de lucru pentru un codor de vorbire de calitate înaltă (telefonică) de 8 kbps. Cerința inițială pentru întârziere a fost redusă în 1991 pentru a reduce complexitatea estimată a codorului.[62] În 1993 și la începutul lui 1994, au fost testate două potențiale codoare. Ambele codoare au corespuns cerințelor de

performanță pentru semnal de vorbire curat la intrare. Oricum, ambele au căzut la testele pentru alte tipuri de intrări, pentru omisiuni de cadre și pentru semnale de vorbire cu fundaluri zgomotoase sau cu muzică. În 1994 s-a ajuns la un acord ca aceste două codoare să fie combinate și ca ambii promotori să colaboreze în vederea producerii unui codor optimizat, care să corespundă tuturor cerințelor. În 1995 testele au fost repetate de patru laboratoare de testare subiectivă în patru limbi diferite. SQEG a conchis că în baza rezultatelor acestor teste, codorul optimizat corespunde tuturor cerințelor de calitate a vorbirii. Ratificarea acestui codor a fost finalizată în noiembrie 1995 ca Recomandarea G.729 a ITU.

Programul de lucru pentru G.723.1 a fost inițiat de ITU în 1993, ca parte a unui grup de standarde care să specifice un videofon cu rată de bit scăzută, pentru utilizarea în rețeaua publică de telefonie PSTN (*Public Switched Toll Network* [NT5]) prin intermediul unui modem de viteză mare. Alte standarde din acest grup includ codorul video, modemul și schema de multiplexare a datelor. Intenția a fost de a selecta unul dintre codoarele de vorbire existente, puse la dispoziție de tehnologia de codare „din stoc". Calitatea trebuia să fie cât se poate de bună în limitele unei complexități nu prea mari și a unei dimensiuni a cadrelor de maxim 30 ms. Un test subiectiv de ascultare a fost realizat pentru 5 codoare propuse, în martie 1994. Două codoare au fost selectate pentru studii ulterioare și promotorii lor au acceptat în final compromisul de a crea un codor cu rate de bit între 5,3 și 6,3 kbps.[63]

Codorul cu rată de bit mai mare a fost echivalat calitativ cu G.726 ADPCM de 32 kbps pentru vorbire curată la intrare. Oricum, calitatea lui pentru vocile de copii, fundal zgomotos sau muzică nu este la fel de bună ca la G.726. Rata de 5,3 kbps a fost inclusă pentru a putea oferi mai multă flexibilitate proiectanților de sisteme. Dacă nu se poate stabili decât o legătură de viteză mică, de 9,6 sau 14,4 kbps, poate fi avantajos să se folosească un codor cu o rată de bit mai joasă, pentru a putea aloca mai mulți biți codorului video.

Codorul de 6,3 kbps folosește o formă de codare multipuls LPC a vorbirii[14], care limitează posibilitățile de poziționare a impulsurilor. Codorul de 5,3 kbps folosește o agendă de coduri (*codebook*) algebrică[23] pentru excitație. De fapt, aceasta este tot o formă de codare multipuls, care limitează posibilitățile de poziționare a impulsurilor. Codorul a fost, de asemenea, menționat ca un candidat pentru aplicații cu voce și date simultane.

Acest codor și G.729 au fost primele codoare ITU specificate prin simularea ANSI C cu virgulă fixă, a codorului și a decodorului. Măsuri de precauție pentru acest lucru au fost incluse la ieșire, pentru a reduce timpul necesar pregătirii specificației. Aprobarea finală a acestui standard de SG15 a fost dată în noiembrie 1995.

Pentru a oferi capacități de codare simultană a datelor și a vorbirii, a fost inițiat un program la solicitarea SG14 pentru un codor de vorbire de complexitate redusă, pentru utilizare cu modemurile V.34. Un calendar foarte strict a fost stabilit, cu finalizare planificată în noiembrie 1995. Complexitatea urma să fie limitată în jurul a 10 MIPS. În același timp, SG15 a cerut o calitate apropiată de telefonia analogică și o bună calitate la utilizarea în tandem, astfel încât codorul să poată fi utilizat pentru diferite scenarii,

[NT5] În alte lucrări prin Public Switched Telephone Network; se înțelege rețeaua cablată de telefonie clasică (fixă).

inclusiv conferințele între două sau mai multe puncte. Pentru a reduce complexitatea standardelor curente ITU, rata de bit maximă a fost crescută la 11,2 kbps. Interpolarea digitală a vorbirii va fi folosită pentru a face rata medie de bit mult mai mică.

Codorul selectat a fost o variantă a Recomandării G.729 a ITU și este cunoscută acum ca Anexa A a acestui standard. Fluxul de biți este același ca la G.729. El folosește o procedură de căutare simplificată a excitației, unele mici simplificări ale analizei LSP și un postfiltru mai simplu. Ca rezultat al acestor simplificări, complexitatea totală a fost redusă la 10,5 MIPS. Memoria RAM a fost redusă la mai puțin de 2 K. Detectorul de activitate vocală și generatorul de zgomot de confort au schemele date de Anexa B la Recomandarea G.729. Performanțele acestui codor sunt ușor mai reduse decât cele ale G.729, datorită acestor simplificări. O caracteristică bună a codorului și decodorului de acest tip este că ele pot interopera cu codorul și decodorul G.729. Mai multe detalii sunt cuprinse în [80].

Tabelul 3.3. Planificarea ITU pentru viitoare standarde

Organizația de standardizare	ITU	ITU
Tipul de codor	Necunoscut	Necunoscut
Planificare		
Data primei testări	Trim 2 / 1996	Trim 1 / 1996
Data finală	1998	1998
Rata de bit	4 kbps	16-32 kbps
Calitatea	Telefonie	Comentariu
Complexitatea	Necunoscută	Vezi nota1
Întârzierea		
Mărimea cadrului	Max. 20ms	Vezi nota 1
Previziune	Max. 5 ms	
Tipul specificației		
Virgulă mobilă	valoarea nu este fixată	Vezi nota 1
Virgulă fixă	Bit fix C	Bit fix C

Nota 1: Codorul de bandă largă are două moduri posibile. Modul A este de întârziere redusă (mărimea cadrului max. 10 ms) și care poate crea o complexitate mai mare. Modul B urma să aibă o complexitate mai redusă (max. 15 MIPS) și poate avea o întârziere de până la 30 ms pe cadru. Se intenționează să se implementeze codorul de complexitate redusă atât în circuite DSP cu virgulă fixă, cât și în procesoare gazdă ale PC-urilor de mare performanță și ale stațiilor de lucru. Ambele codoare au ca țintă comună o calitate care, la 16 kbps, să fie egală sau mai bună decât G.722 pe 48 kbps, la 24 kbps, să fie cel puțin egală cu G.722 la 56 kbps, iar la 32 kbps să fie cel puțin egală cu G.722 la 64 kbps.

3.3. *Programele ITU pentru standarde viitoare*

Tabelul 3.3 descrie agenda curentă de lucru a ITU pentru viitoarele recomandări privind codoarele de vorbire. În 1997 se știa că unul dintre ele urma să fie un codor cu predicție liniară cu analiză-prin-sinteză (LPAS), derivat ca o extensie a unuia dintre codoarele descrise în capitolul 3.2. Toate celelalte codoare vor fi proiecte cu totul noi.

Un efort de definire a unui codor de 4 kbps a fost întocmit după toate formele în mai 1994, când SG15 a aprobat Termenii de Referință în cadrul unei ședințe.[64] Aplicațiile avute în vedere sunt telefonia vizuală de viteză de bit foarte mică, comunicațiile interpersonale și sistemul de telefonie mobilă prin satelit. La testele inițiale din mai

1996, nici unul dintre codoare nu a acoperit un număr suficient de cerințe pentru a fi admis. O nouă testare a fost programată în martie 1997.

O încercare de codare a vorbirii de 7 kHz a fost formulată în februarie 1995 când SG15 a aprobat Termenii de Referință. Termenii de Referință provizorii sunt cuprinși în [65]. Aplicațiile vizate sunt telefonia de bandă largă prin ISDN, videotelefonia ISDN și videoconferințele cu acces la vitezele de bază, aplicațiile de tip pachet de bandă largă ISDN sau ATM, circuite de multiplexare digitale, aplicații PSTN de legături punct-la-punct (prin modem) și serviciul de mesaje. Modul A este dedicat uzului general și va suporta codări multiple, care necesită posibilități bune de tandemizare și întârzieri reduse. Modul B este opțiunea de complexitate redusă și întârziere mare și poate fi folosită în aplicații punct-la-punct. Se dorește ca modul B să aibă o complexitate atât de scăzută încât să poată opera pe stații de lucru de nivel înalt și PC-uri, într-o mică fracțiune (15-20%) de timp real.

3.4. *Standarde de codare a vorbirii pentru telefonie celulară digitală*

Tabelul 3.4. descrie prima și a doua generație de codoare de vorbire standardizate pentru telefonia celulară digitală.

Tabelul 3.4. Prima și a doua generație de codoare de vorbire pentru telefonia celulară digitală

Organizația de standardizare	CEPT	ETSI	TIA	TIA	RCR	RCR
Numele standardului	GSM	GSM ½ rate	IS-54	IS-96	PDC	PDC ?
Tipul de codor	RPE-LTP	VSELP	VSELP	CELP	VSELP	PSI-CELP
Data	1987	1994	1989	1993	1990	1993
Rata de bit	13 kbps	5,6 kbps	7,95 kbps	0,8-8,5 kbps	6,7 kbps	3,45 kbps
Calitatea	< telefonie analogică	=RPE-LTP	=RPE-LTP	<RPE-LTP	<RPE-LTP	=viteză max.
Complexitatea						
MIPS	4,5	30	20	20	20	48 (est)
RAM	1 K	4 K	2 K	2 K	2 K	4 K
Întârzierea						
Mărimea cadrului	20 ms	20 ms	20 ms	20 ms	20 ms	40 ms
Previziune	0	5 ms	5 ms	5 ms	5 ms	10 ms
Tipul specificației						
Virgula fixă	Bit fix	Bit fix C	„flux biți"	„flux biți"	„flux biți"	„flux biți"

Prima generație de codoare a asigurat o calitate adecvată. Două codoare din a doua generație sunt așa numite „la jumătate de viteză" (*half-rate coders*) și au fost introduse pentru a mări capacitatea industriei celularelor, care a cunoscut o mare dezvoltare. O a treia generație le va urma în curând pentru a aduce calitatea vorbirii în serviciile digitale avansate la nivelul actual al telefoniei curente pe linii cablate.

Codorul RPE-LTP a fost standardizat de GSM (*Group Special Mobile*[NT6]) din cadrul CEPT în 1987 pentru telefonie digitală celulară pan-Europeană.[66] RPE-LTP vine de la *Regular Pulse Excitation with Long-Term Prediction* (excitație cu impulsuri regulate și

[NT6] Denumirea oficială a sistemului GSM a devenit ulterior Global System for Mobile Communications.

predicție pe termen lung). Versiunea finală a codorului a fost un compromis între codorul RPE propus de Philips[19] și un codor multipuls LPC propus de IBM France. Canalul GSM de capacitate maximă suportă 22,8 kbps. Completarea de 9,8 kbps este folosită la codarea canalului, pentru a proteja codorul de erori de bit ale canalului radio. Detectarea activității vocale și a discontinuităților de transmisie sunt incluse în acest standard. În plus față de telefonia digitală celulară, acest codor a fost de la început folosit pentru multe aplicații suplimentare, cum ar fi mesajele, datorită complexității sale reduse. În ciuda ratei sale de bit relativ mare, el a fost recomandat pentru aplicații de voce și date simultane.

Codorul GSM la jumătate de viteză a fost standardizat de TCH-HS (succesor al GSM) în scopul dublării capacității sistemului celular GSM. Codorul este de tip VSELP de 5,6 kbps, implementat de Motorola.[23] Un procent mai mare de biți de canal sunt folosiți pentru protecția la erori, deoarece canalul la jumătate din capacitate are o diversitate mai mică decât cel la capacitatea întreagă. Performanțele generale măsurate l-au dovedit comparabil cu RPE-LTP, cu excepția unor semnale cu fundal zgomotos.

La sfârșitul lui 1995, TCH-HS a standardizat un succesor al codorului de viteză maximă RPE-LTP. Este cunoscut ca US-1. El folosește metoda de excitație ACELP. Ca și codorul original GSM, operează la 13 kbps. El asigură calitate comparabilă cu telefonia cablată pentru semnale curate de vorbire la intrare.

Codorul IS-54 a fost standardizat de TIA (*Telecommunications Industry Association*) cu telefonia digitală celulară din America de Nord, cu acces multiplu cu divizare în timp (TDMA = *Time Division Multiple Access*). Este o componentă a *Interim Standard 54* (IS-54). A fost propus de Motorola[67] și ales pe baza testelor de ascultare subiective în 1989. Calitatea acestui codor și a RPE-LTP diferă întrucâtva, datorită caracterului distorsiunilor, dar au obținut aproape același MOS la testele de ascultare subiective. El nu a fost testat inițial în vederea tandemizării sau în condiții de zgomot de fundal și nu dă rezultate bune în aceste condiții. IS-54 nu are o specificație asupra poziției virgulei. Implementarea trebuie să se conformeze numai specificației fluxului de biți. Oricum, TIA are o procedură de calificare, IS-85, pentru a verifica dacă performanțele unei implementări sunt destul de bune pentru a fi folosite în telefonia digitală celulară.[68] În plus, Motorola a furnizat un program cu virgulă mobilă în C pentru versiunea sa de codor, pe care producătorii o pot folosi ca ghid.

Prin finalizarea Recomandării G.729 a ITU în 1995, s-au făcut presiuni pentru a înlocui IS-54 VSELP cu un codor de vorbire de calitate telefonică pe 8 kbps. Aceasta deoarece pe canalele curate, calitatea IS-54 VSELP este mai slabă decât a transmisiei MF analogice. Pentru un timp, TIA a susținut utilizarea G.729, dar în final a decis să selecteze o variantă standardizată acum în IS-641. Codorul IS-641 folosește de fapt aceeași secvență de excitație ca și G.729. Diferența principală constă în faptul că operează cu cadre de 20 ms (fiecare sub-cadru este încă de 5 ms). Economia constă în cuantizarea informației LPC numai la fiecare 20 ms, în loc de fiecare 10 ms. Prin urmare, rata de bit poate fi redusă de la 7,9 kbps la 7,4 kbps cu pierderi mici sau nule de calitate. Biții rămași pot fi folosiți pentru protecția suplimentară la erorile din canal. În comparații neoficiale de ascultare între un sistem bazat pe G.729 și un sistem bazat pe IS-641, se pare că protecția suplimentară a canaluluii conduce la îmbunătățirea comportării terminalelor care se deplasează rapid în zone de fading. Pentru variații lente ale fadingului, rezultatele nu indică îmbunătățiri semnificative. Aceasta era de așteptat, deoarece ștergerea cadrelor este degradarea dominantă la viteze reduse. Într-un

canal curat, IS-641 asigură performanțe aproape egale cu G.729, în aproape orice condiții ale semnalelor de intrare sau de tandem. IS-641 va fi folosit cu sistemele TDMA digitale IS-36 și în timp îl va înlocui pe IS-54 VSELP.

Codorul IS-96 a fost standardizat de TIA pentru telefonia celulară digitală cu acces multiplu cu divizare de cod (CDMA = *Code Division Multiple Access*) din America de Nord.[69] Este parte componentă a IS-96 și este folosit în sistemul specificat de IS-95. Pentru sistemele CDMA, capacitatea este caracteristica cea mai atractivă, și pentru a atinge această capacitate, în pauzele de vorbire, rata canalului trebuie mult redusă. IS-96 QCELP este un codor cu rată de bit variabilă, care folosește interpolarea digitală a vorbirii pentru a obține această reducere a vitezei. El operează la 8,5 kbps pe timpul prezenței sunetelor vorbirii. În pauzele de vorbire de pe canal, viteza scade până la 0,8 kbps. La această rată, el furnizează numai date statistice despre zgomotul de fond. Aceste două rate sunt cel mai des folosite în timpul operării unui IS-96, deși codorul mai face transmisii și cu două viteze intermediare, de 2 și 4 kbps. Procedura de validare a acestui codor este similară cu cea pentru IS-85.[70]

Interpolarea digitală a vorbirii lucrează cel mai bine dacă zgomotul de fond este complet mascat de sunetele vorbirii și poate fi înlocuit de zgomot sintetic în celelate momente. Folosirea interpolării digitale a vorbirii în telefonia celulară micșorează calitatea vorbirii, mai ales în condiții de fond foarte zgomotos. Dacă nivelul zgomotului de fond este ridicat, el poate declanșa viteza maximă de bit a codorului în intervalele de pauză ale vorbirii și poate să nu fie mascat de vorbire. Acest codor are dificultăți de operare în astfel de condiții. În condiții ideale, cu interpolarea digitală a vorbirii dezactivată, codorul este încă sub calitatea unui IS-54 VSELP. În consecință, TIA l-a vizat pentru a fi înlocuit. În 1995 s-a desfășurat un test subiectiv pentru a determina dacă un alt codor poate oferi o calitate mai bună decât IS-96. Mai multe codoare au reușit. Standardul final este încă în completare. În cadrul TIA, acest demers este numit întrecerea pentru codorul îmbunătățit cu rată variabilă EVRC (*Enhanced Variable Rate Coder*).

Codorul cu viteză întreagă PDC (*Personal Digital Cellular*) a fost standardizat de RCR (*Research and Development Center for Radio Systems*), pentru serviciul de telefonie digitală celulară TDMA din Japonia, sub codul RCR SDT-27B. Codorul a fost propus de Motorola și este similar cu IS-54 VSELP.[58] Diferența principală constă în aceea că în loc de două matrice de codare (*codebook*) de excitație cu sumă vectorială, există numai una. Cele două matrice de codare din IS-54 au fiecare câte 7 vectori de bază, în timp ce cea din PDC-VSELP are 9 vectori de bază. În termeni de complexitate a căutării, cele două codoare VSELP sunt aproape egale. O comparație a performanțelor primelor trei standarde digitale celulare (RPE-LTP, IS54 și PDC-VSELP) a fost publicată în [71]. Concluzia a fost că PDC-VSELP este mai puțin performant decât celelalte două.

Codorul PDC la jumătate de viteză a fost standardizat de RCR pentru a dubla capacitatea sistemului japonez TDMA PDC.[72,73] El a fost creat de NTT. PSI-CELP (*Pitch Synchronous Innovation CELP*) folosește matrice de codare fixe care sunt modificate în funcție de tonuri, pentru a îmbunătăți calitatea vorbirii la aceste rate de bit foarte scăzute. Dacă durata tonului este mai mică decât dimensiunea cadrului, atunci toți vectorii din matricea de codare fixă, pentru acel cadru, sunt făcuți periodici. Dimensiunea cadrului este de 80 de eșantioane. Să presupunem că tonul este de 50. Atunci, în loc să folosească ultimele 30 de eșantioane ale fiecărui vector în matricea de codare, primele 30 de eșantioane ale acelui vector se repetă. Dacă tonul este de 25, primele 25 de eșantioane s-ar repeta de 3 ori și

încă o dată primele 5. Trebuind să creeze matricele de codare în RAM înainte de a le căuta, efortul de calcul pentru căutare trebuie să fie foarte mare şi poate, de asemenea, să necesite mai mult RAM. În prezent, acest codor pare să fie cel mai complex standardizat până acum. O ultimă caracteristică interesantă a acestui codor este că are un pre-procesor de zgomot de fond ca parte a standardului. Când acesta sesizează că zgomotul de fond a depăşit un anumit prag, pre-procesorul încearcă să îmbunătăţească calitatea vorbirii.

3.5. *Codoare de vorbire pentru sisteme bazate pe satelit*

Tabelul 3.5. descrie trei dintre codoarele care au fost standardizate pentru telefonia fără fir, bazată pe satelit. Telefonia bazată pe satelit este abia în stadii incipiente şi un număr de consorţii intenţionează să construiască sisteme adiţionale pentru a oferi o acoperire a întregii lumi, astfel încât clienţii să poată da şi primi telefoane oriunde şi oricând, la preţuri accesibile.

Tabelul 3.5. Codoare de vorbire pentru sisteme bazate pe satelit

Organizaţia de standardizare	AEEC	Inmarsat
Numele standardului	AMTS	M
Tipul de codor	Multipulse LPC	IMBE
Data	1988	1990
Rata de bit	9,6 / 19,2 kbps	4,15 / 6,4 kbps
Calitatea	< GSM	< GSM
Complexitatea		
MIPS	8 (estimată)	7
RAM	2 K	< 2 K
Întârzierea		
Mărimea cadrului	20 ms	20 ms
Tipul specificaţiei	flux de biţi	flux de biţi

Notă: Ratele de bit specificate primele sunt pentru codarea vorbirii, iar următoarele pentru întregul canal.

Deoarece lăţimea de bandă a canalului de satelit este o resursă limitată, păstrarea ratei de bit cât se poate de redusă este extrem de importantă. În viitor, telefonia prin satelit va fi o aplicaţie care va tinde să ducă rata de bit a codoarelor de vorbire încă şi mai jos. Ca toate canalele radio, fluxul de biţi este supus unor posibile erori aleatoare şi fadingului. Aceasta necesită un accent pus pe rezistenţa la erori. Protecţia la erorile canalului este în mod uzual încorporată în aceste sisteme.

Standardul serviciului de telefonie mobilă aviatică AMTS (*Airborne Mobile Telephony Service*) a fost creat în 1988. *Airlines Electronics Engineering Committee* a testat câteva codoare de vorbire şi a ales un codor de 9,6 kbps, multipuls LPC. Fluxul de biţi de 9,6 kbps codat pentru canal conduce la un total de 19,2 kbps. Implementarea iniţială a folosit un procesor de prelucrare digitală a semnalelor AT&T DSP32, care este un dispozitiv cu virgulă mobilă. Complexitatea listată în tabelul 3.5 de mai sus este bazată pe o estimare a numărului echivalent de MIPS pentru virgulă fixă. Sistemul comunică între un satelit geostaţionar şi reţeaua telefonică terestră cablată. Acest standard este folosit de liniile aeriene care oferă serviciul de aerofonie (*Airphone*).

Codorul Inmarsat M a fost standardizat de Inmarsat (*International Maritime Satellite Corporation*) în 1990 pentru comunicarea în întreaga lume între navele maritime şi

țărm via satelit (Inmarsat-M) și de către Optus / Aussat pentru telefonia mobilă prin satelit. Viteza totală a canalului este de 6,4 kbps. Codorul de vorbire folosește numai 4,15 kbps iar biții rămași sunt folosiți la codarea canalului. Principiile fundamentale ale codorului IMBE (*Improved Multi-Band Excitation*) sunt descrise în [49], [50] și au fost, de asemenea, descrise în capitolul 2.10. O nouă versiune a codorului IMBE a fost creată pentru standardul Inmarsat mini-M, care a fost finalizat în 1994. Pentru acest standard, viteza canalului este de 4,8 kbps. Detaliile complete ale acestui codor nu sunt încă disponibile public. O versiune a codorului IMBE a fost creată și pentru canalele civile destinate urgențelor, cum ar fi canalele pentru urgențe ale poliției. Această variantă a fost adoptată ca standard de către TIA pentru America de Nord, ca rezultat al testărilor efectuate de APCO/NASTD/Fed Project 25, pentru utilizarea în sistemele radio digitale mobile terestre. Acest codor a fost implementat de DVSI în numeroase circuite DSP (incluzând toți producătorii majori), atât cu virgulă fixă cât și cu virgulă mobilă.

3.6. *Standarde de vorbire securizată (cifrată): FS1015, FS1016 și un nou codor pe 2,4 kbps*

Tabelul 3.6. prezintă informații despre trei standarde de codoare de voce securizată. Două sunt standarde existente, cel de-al treilea descrie un codor propus de guvernul Statelor Unite.

FS1015 este un standard federal al S.U.A. pentru un vocoder de 2,4 kbps, care a fost creat într-un timp îndelungat, începând de la sfârșitul anilor '70. A fost standardizat de Departamentul Apărării (DoD = *Department of Defense*) și mai târziu de NATO, înainte de a deveni standard federal în 1984. A fost destinat terminalelor de voce protejată. El nu produce sunete de vorbire naturale, dar în decursul anilor inteligibilitatea lui a fost mult îmbunătățită printr-o serie de schimbări atât în codor cât și în decodor. În mod remarcabil, aceste schimbări nu au necesitat niciodată modificarea fluxului de biți. În prezent, inteligibilitatea lui FS1015 pentru semnal curat de vorbire la intrare, de bandă telefonică, este aproape identică cu a materialului sursă așa cum au dovedit-o măsurătorile testelor de diagnosticare cu rimă (DRT = *Diagnose Rhyme Test*).[74] Mai recent, o versiune a FS1015 de cuantizare vectorială de 800 bps a fost standardizată de NATO.[75,65]

Tabelul 3.6. Standarde de codare a vorbirii pentru telefonia protejată

Organizația de standardizare	U.S. DoD	U.S. DoD	U.S. DoD
Numele standardului	FS1015	FS1016	new 2,4 kbps
Tipul de codor	LPC vocoder	CELP	Bazat pe modele
Data	1984	1991	1996
Rata de bit	2,4 kbps	4,8 kbps	2,4 kbps
Calitatea	DRT înaltă	< IS-54	≥ FS1016
Complexitatea			
MIPS	20	19	<80 (vezi nota)
RAM	2 K	1,5 K	valoare nefixată
Întârziere			
Mărimea cadrului	22,5 ms	30 ms	
Previziunea	90 ms	7,5 ms	
Tipul specificației	flux de biți	flux de biți	flux de biți

Notă: Ținta este de 40 MIPS pentru virgulă fixă și 80 MIPS pentru virgulă mobilă.

FS1016 este rezultatul unui proiect preluat de DoD pentru a face mai natural caracterul sunetului la telefonul securizat III (STU-3), prin introducerea tehnologiei de modem de 4,8 kbps. DoD a supravegheat tehnologia codoarelor de vorbire de 4,8 kbps disponibile în 1988 și 1989. El a selectat codorul bazat pe CELP, care folosește așa-numita matrice de codare ternară, ceea ce înseamnă că amplitudinile excitației pot fi –1, 0 și 1 înainte de scalarea printr-o amplificare a sub-cadrelor.[77] Aceasta permite o căutare mai ușoară în matricea de codare. FS1016 păstrează cu mult mai bine naturalețea vorbirii decât FS1015, dar vorbirea păstrează încă multe artefacte (produse spectrale false) iar calitatea este substanțial sub cea a codoarelor celulare ca GSM sau IS-54.

Următorul codor care urmează a fi standardizat de DoD este un codor nou de 2,4 kbps, care le va înlocui atât pe FS1015 cât și pe FS1016. Un proiect de trei ani a fost inițiat în 1993, care a culminat cu selectarea unui nou standard în 1996. Necesitatea acestui codor provine din lipsa unui număr suficient de canale de satelit pentru viteza de 4,8 kbps. Prin înlocuirea cu un codor de numai 2,4 kbps, numărul canalelor care pot să îl preia se dublează. Calitatea dorită pentru acest codor este mai mare sau egală cu calitatea și inteligibilitatea lui FS1016 pentru cea mai mare parte a scenariilor. Multe dintre scenarii includ severe zgomote de fond și condiții de canal foarte zgomotos. La 2,4 kbps nu sunt suficienți biți disponibili pentru o codare explicită a canalului, deci codorul de vorbire trebuie să fie el însuși prin proiectare și construcție imun față de condițiile canalului. Fondul zgomotos s-a dovedit a fi o dificultate pentru vocoderele care iau decizia „cu voce / fără voce", indiferent dacă decizia se ia pentru întreaga bandă sau numai pentru benzi izolate. Teste subiective s-au făcut în 1993 și 1994 pentru diferite variante de software ale potențialelor codoare și o evaluare hardware în timp real a început în a doua parte a lui 1995, pentru a selecta cel mai bun candidat. În testele din 1994 a existat un candidat pe 2,4 kbps care a corespuns sau a depășit toate cerințele.[78,79] La testele hardware au corespuns sau au depășit cerințele trei codoare. A fost ales codorul MELP (*Mixed Excitation Linear Prediction*) pe baza performanțelor și a complexității. Detalii despre acest codor pot fi găsite în [81].

3.7. *Performanțe*

Figura 12 este inserată pentru a da o impresie asupra performanțelor relative în condiții de vorbire clară la intrare, a majorității codoarelor standard care au fost descrise până aici. Nu s-a realizat niciodată până în prezent un test subiectiv care să includă toate codoarele de mai sus. Figura 12 se bazează pe performanțele relative ale acestor codoare, de-a lungul a numeroase teste care au fost raportate. În cazul codoarelor care nu reprezintă încă un standard, performanțele lor sunt marcate cu un cerculeț. Axa verticală a figurii 12 arată calitatea individuală aproximativă pentru un semnal curat de vorbire la intrare. Axa orizontală este o scară logaritmică a ratei de bit. Figura 12 include numai codoare de vorbire de bandă telefonică. Codoarele de vorbire de 7 kHz au fost omise. Figura 13 compară complexitatea măsurată în MIPS și RAM pentru implementarea cu un DSP cu virgulă fixă, pentru cele mai multe dintre aceleași codoare standard. Axa orizontală este în RAM iar axa verticală este în MIPS.

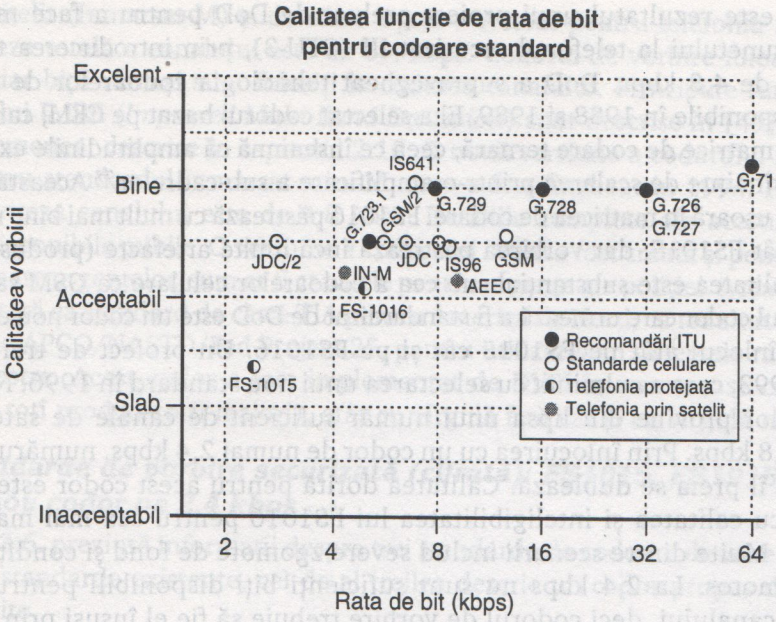


Figura 12. Calitatea relativă pentru o singură codare a unui semnal curat de vorbire, de către o varietate de codoare standard, ca funcție de rata de bit. (Notă: această figură este orientativă, bazată pe o serie de rezultate ale unor teste diferite. Ea nu reprezintă rezultatul nici unui test subiectiv particular.)

4. Implementări ale codoarelor de vorbire

Tradițional, codoarele de vorbire sunt create întâi prin simulări și apoi implementate într-un procesor în timp real. Pe parcursul ultimilor 15 ani, procesoarele în timp real au fost mai mult sau mai puțin sinonime cu circuitele integrate DSP (*Digital Signal Processor*). Pe urmă, ele au putut fi realizate într-un circuit integrat pe scară foarte largă (VLSI) destinat unei aplicații speciale (*Special Purpose VLSI*). Un astfel de circuit se mai numește ASIC (*Application Specific Integrated Circuit*). Cum cele mai multe procesoare gazdă ale PC-urilor și ale stațiilor de lucru devin tot mai puternice, codoare de vorbire sunt și ele realizate prin software pe aceste procesoare. Astfel, procesorul vizat poate fi un cip DSP sau un procesor gazdă CPU sau un ASIC sau toate trei la un loc. Procesorul avut în vedere impune restricții asupra complexității codorului de vorbire. În acest capitol vom revedea diferiții pași necesari implementării în timp real sau realizării.

Primul pas în procesul de implementare este crearea codorului de vorbire. Aceasta se face cel mai adesea prin simulări pe un calculator de uz general, cum ar fi un PC performant sau o stație de lucru. Este de dorit ca simularea să fie scrisă într-un limbaj de nivel înalt, cum sunt C sau C++. Codoarele de vorbire mai recente sunt destul de complexe și pot necesita mii sau zeci de mii de linii de program. Foarte adesea aceste programe lungi suferă de problema firească a programelor lungi. Erorile de programare (*bugs*) trec nedetectate deoarece sunt mascate de numărul mare de linii de program. Cu cât codul sursă crește, el devine mai ineficient în utilizarea memoriei și prin numărul de instrucțiuni.

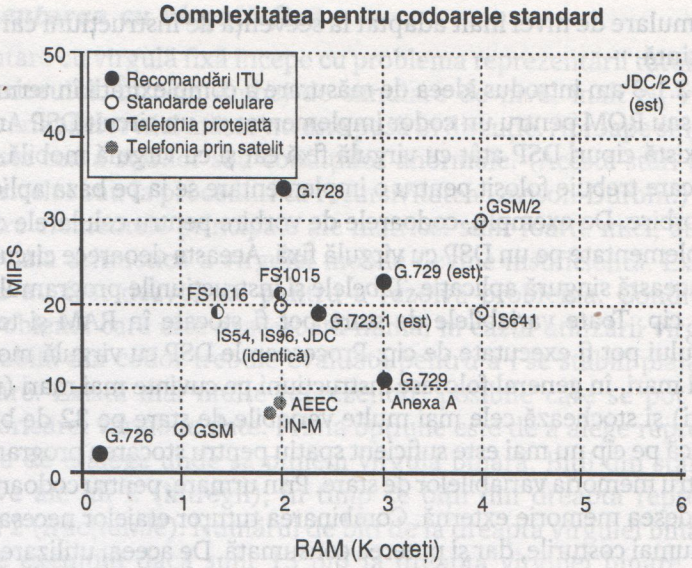


Figura 13. Complexitatea relativă a diferitelor codoare standard, exprimată în MIPS și consum de RAM (valorile sunt aproximative)

Mai recent, cercetătorii au folosit programe ca Matlab® care oferă două avantaje importante în fața acestor probleme. În primul rând, aceste limbaje conțin funcții primitive puternice care pot fi folosite pentru a construi o reprezentare compactă a codorului de vorbire. De exemplu, Matlab conține inversarea matriceală ca funcție primitivă. Această compactizare este dorită deoarece îl ajută pe creator să păstreze un control mai bun al codurilor sursă. Utilizarea funcțiilor primitive puternice înseamnă și că cel care creează algoritmul nu trebuie să se îngrijoreze de apariția, din greșeală, a erorilor de programare în cadrul acestor funcții. (Un avantaj suplimentar al unor astfel de funcții primitive este faptul că ele scutesc creatorul algoritmului de irosirea timpului cu aceste detalii de joasă complexitate ale programului.) Un al doilea avantaj este faptul că aceste programe furnizează, de asemenea, posibilități grafice deosebit de puternice, care permit autorului algoritmului să vizualizeze modul în care sunt procesate datele. Adeseori aceste posibilități sunt utilizate și au ca rezultat noi interpretări în cadrul operațiilor algoritmului. Această combinație între notația compactă utilizând eficient funcții puternice și posibilități grafice, permit proiectanților să lucreze eficient în etapele de început ale creării unui codor. Proiectantul poate să încerce mai multe idei, poate analiza munca mai bine cu ajutorul facilităților grafice, iar codurile sursă rezultate vor fi mai puțin susceptibile de a conține erori și vor fi mai bine organizate.

O limitare a acestor programe este faptul că adesea nu permit simularea completă a diferitelor grade de precizie. Pentru a evita erorile de precizie, ele reprezintă de obicei toate variabilele în dublă precizie cu virgulă mobilă. Macrofuncțiile acestor programe există numai în acest format și nu permit proiectantului să simuleze alte reprezentări. Noile versiuni ale acestor programe încep să permită și reprezentări în formate adiționale.

Crearea unui codor este numai un prim pas în procesul de realizare. O dată proiectul inițial creat, codorul trebuie testat pentru a determina dacă răspunde tuturor cerințelor. Când pragul acesta a fost atins, începe al doilea pas în vederea realizării: conversia la

un limbaj de simulare de nivel înalt adaptat la secvența de instrucțiuni care va fi realizată de procesorul țintă.

În capitolul 2.1.3 am introdus ideea de măsurare a complexității în termeni de utilizare de MIPS, RAM sau ROM pentru un codor implementat cu un circuit DSP. Am menționat de asemenea că există cipuri DSP atât cu virgulă fixă cât și cu virgulă mobilă. Decizia asupra tipului de DSP care trebuie folosit pentru o implementare se ia pe baza aplicației primare a codorului de vorbire. De exemplu, codoarele de vorbire pentru celularele digitale sunt cel mai adesea implementate pe un DSP cu virgulă fixă. Aceasta deoarece circuitul este utilizat numai pentru această singură aplicație. Tabelele și instrucțiunile programului pot fi stocate în ROM-ul din cip. Toate variabilele de stare pot fi stocate în RAM și toate procesările necesare codorului pot fi executate de cip. Procesoarele DSP cu virgulă mobilă au CPU de dimensiuni mai mari, în general folosesc instrucțiuni pe cuvinte mai mari (uzual 32 de biți în loc de 16 biți) și stochează cele mai multe variabile de stare pe 32 de biți în loc de 16. Rezultatul este că pe cip nu mai este suficient spațiu pentru stocarea programului, a tuturor tabelelor și pentru memoria variabilelor de stare. Prin urmare, pentru codoare mai complexe este necesară adesea memorie externă. Combinarea tuturor etajelor necesare pe un singur cip reduce nu numai costurile, dar și puterea consumată. De aceea, utilizarea procesoarelor DSP cu virgulă fixă este preferată pentru aplicații dedicate.

O nouă opțiune este de a proiecta un circuit integrat dedicat pentru o aplicație specifică (ASIC), codorul de vorbire. Aceasta este o opțiune atractivă pentru codoarele de vorbire care au complexitate relativ redusă. În plus, folosirea unui ASIC trebuie să ofere un avantaj față de utilizarea unui cip DSP cu virgulă fixă. Cum tehnologia VLSI continuă să evolueze, distincția între cipurile DSP cu virgulă fixă și ASIC începe să se estompeze. Unora dintre proiectanții de DSP li s-a oferit posibilitatea de a li se livra părți pentru cipul lor. De exemplu, o companie poate să aleagă unitatea centrală și să o adauge la propria bibliotecă de CAD (*Computer Aided Design*), împreună cu descrierea perifericelor, memoriile RAM și ROM și apoi să proiecteze la cerere un DSP pentru o aplicație particulară. Avantajul este că părțile unui DSP standard, care nu sunt strict necesare aplicației, sunt eliminate. Ele nu mai trebuie să fie plătite. Și nu mai consumă putere. Cipul rezultat va fi mai mic și mai ieftin. Este el un DSP sau un ASIC? Din punctul de vedere al cercetătorului sau al celui care dezvoltă un codor de vorbire, acest lucru nu mai este important. Pașii necesari implementării pe un ASIC par foarte asemănători cu cei pentru un DSP cu virgulă fixă.

Există însă cazuri în care un singur procesor este utilizat pentru mai multe aplicații. Ne putem imagina ca pe parcursul vieții unui produs să i se poată adăuga aplicații sau să se modifice aplicații existente. În acest caz, este preferat un DSP cu virgulă mobilă deoarece este mai ușor de programat. El asigură o platformă flexibilă care suportă un număr mare de aplicații. O altă opțiune care poate fi luată în considerare este dacă se poate realiza un codor de vorbire folosind numai o parte din puterea de calcul a unui procesor gazdă. Această opțiune este atractivă pentru adăugarea capacităților de comunicare vocală la stații de lucru sau la calculatoare personale la costuri suplimentare reduse atât pentru producător cât și pentru cumpărător. În prezent, codoarele care pot să ruleze pe un procesor gazdă sunt limitate la complexități destul de reduse, dar cum unitățile centrale de calcul ale acestor terminale continuă să se dezvolte în capacitate, această opțiune va deveni din ce în ce mai comună.

În capitolul următor, vor fi discutate etapele și problemele asociate fiecărui tip de implementare.

4.1. *Implementarea cu virgulă fixă*

O implementare cu virgulă fixă începe cu problema reprezentării tuturor variabilelor de stare ale codorului. Într-un limbaj de simulare de nivel înalt cu virgulă mobilă, reprezentarea variabilelor nici măcar nu este pusă în discuție. Aproape singura problemă care se pune este cea a stărilor sau condițiilor anormale. (Aceste stări apar adesea la inversarea matricelor sau la proceduri ca recursivitatea Levison-Durbin.) Dacă unul sau mai multe dintre elementele principale ale matricei sunt foarte mici, atunci în timpul procedurii, precizia aritmetică a virgulei mobile devine insuficientă. Există un număr de tehnici care au fost dezvoltate pentru a rezolva problema. Condițiile anormale reprezintă o problemă care este exacerbată numai în cazul utilizării virgulei fixe.

Fiecare variabilă din codor trebuie evaluată pentru a i se stabili plaja de variație și precizia necesare. Există mai multe reprezentări posibile care se pot utiliza pentru reprezentarea oricărei variabile date. Prima opțiune este de a alege reprezentarea fixă. Avem libertatea de a alege unde să punem virgula binară. Biții din stânga reprezintă puterile pozitive ale lui 2 (întregii), în timp ce biții din dreapta reprezintă puterile negative ale lui 2 (fracțiunile). Numărul de biți de la dreapta virgulei binare se numește valoarea Q. De exemplu dacă sunt 15 biți la dreapta virgulei binare, atunci putem reprezenta fracțiuni cu 15 biți de precizie, dar plaja este restrânsă la [-1, +1), deoarece bitul cel mai din stânga reprezintă semnul. Această reprezentare este numită Q15.

O altă reprezentare este numită virgulă scalară mobilă. În acest caz se folosesc două cuvinte de memorie. Unul reprezintă mantisa, memorată în format Q15. Al doilea reprezintă exponentul. Aceasta oferă o plajă dinamică mai mare decât cele mai multe reprezentări cu virgulă mobilă pe 32 de biți, de vreme ce folosește adesea numai 8 biți pentru exponent. Oricum, Q15 pentru mantisă dă o precizie mai mică.

Dacă variabila este parte a unui vector, putem utiliza declarații de bloc. Din nou, toate elementele pot fi reprezentate ca aceeași valoare fixă Q. Sau, pot să aibă toate același exponent și fiecare să fie reprezentată de propria mantisă. Numim aceasta o virgulă mobilă de bloc (*block floating point*). Pentru vectori mari, se folosește adesea un alt aranjament încă și mai complex. Vectorul este divizat în segmente și fiecare are propria reprezentare cu virgulă mobilă de bloc. Numim aceasta virgulă mobilă de bloc segmentat (*segmented block floating point*).

Problema cu aceste reprezentări diferite este că ele complică implementarea codorului. Programatorul ansamblului DSP pentru codor trebuie să țină minte fiecare dintre reprezentările asociate variabilelor. Pentru fiecare operație aritmetică trebuie pusă întrebarea „care sunt reprezentările operatorilor și rezultatului?" Ar putea fi necesar să re-normalizăm rezultatul pentru o altă reprezentare. Toate acestea cer mai multe instrucțiuni decât dacă s-ar programa în virgulă mobilă. Ca urmare a acestor lucruri și a altor particularități ale arhitecturii cipurilor DSP cu virgulă fixă, o implementare în virgulă fixă necesită, de obicei, de două până la de trei ori mai multe MIPS decât o implementare cu virgulă mobilă. În schimb, cipurile DSP cu virgulă fixă merg la viteze de două, trei ori mai mari decât cipurile DSP cu virgulă mobilă, deoarece unitățile lor aritmetice sunt mai puțin complicate.

Uneori chiar natura algoritmilor din codorul de vorbire trebuie schimbată datorită problemelor de reprezentare. Este posibil să existe o soluție pentru păstrarea aceleiași structuri ca în cazul virgulei mobile, dar să necesite prea multe locații de memorie

și/sau prea multe MIPS. În această situație, ceea ce trebuie făcut este să se caute o cale diferită de a îndeplini același obiectiv, o cale mai simplu de implementat în virgulă fixă. Un bun exemplu îl reprezintă metoda utilizată pentru a calcula funcția de autocorelație pentru analiza LPC în codorul de vorbire G.728. În versiunea sa originală, cu virgulă mobilă, s-a folosit o tehnică cu fereastră recursivă.[23] Când s-a luat în considerație implementarea cu virgulă fixă, s-a constatat că era necesară dubla precizie pentru a implementa fereastra recursivă. În locul ei s-a folosit o tehnică hibridă de fereastră, inedită, care a fost creată pentru a realiza același lucru. Noua tehnică nu numai că cerea o precizie mai scăzută, dar putea să fie implementată mai eficient atât în virgulă mobilă cât și în virgulă fixă. Acesta a fost un exemplu în care, forțați fiind de a schimba procedura datorită unei lipse de precizie a procesorului, a rezultat o îmbunătățire a codorului.

De obicei, reprezentările în virgulă fixă sunt evaluate prin substituirea lor în reprezentări pe virgulă mobilă într-un limbaj de programare de nivel înalt și observând efectele asupra performanțelor codorului. Aceasta poate implica procesarea unor semnale de vorbire și efectuarea de măsurători obiective ca raportul Semnal/Zgomot, sau ascultarea semnalului de vorbire de ieșire. În funcție de variabile, măsurarea S/N poate fi mai mult sau mai puțin utilă decât simpla ascultare. Pentru un codor de vorbire având deja 2000 de variabile, acesta poate fi un proces îndelungat. De multe ori acest lucru se realizează pe baza experienței prealabile a unui expert în codoare de vorbire.

O dată determinată reprezentarea variabilelor, codorul poate fi scris într-un nou program de simulare cu virgulă fixă. Astăzi este avantajos dacă această simulare se scrie în C sau C++. Acest limbaj de nivel înalt servește acum de model pentru implementarea în DSP. Dacă s-a avut grijă să se copieze aritmetica din DSP, atunci simularea în limbajul de nivel înalt și implementarea în DSP vor produce același rezultat față de același semnal de intrare.

Motivul pentru care C și C++ sunt avantajoase este acela că cei mai mulți producători de DSP au acum compilatoare disponibile pentru cipurile lor DSP cu virgulă fixă. Aceste compilatoare nu produc în mod particular programe sursă asamblate foarte eficient, dar programele sursă pot să reproducă rezultatele simulării prin limbajul de nivel înalt. Este apoi rolul programatorului în limbaj de asamblare de a rescrie codorul de vorbire, modul cu modul, pentru a-l face mai eficient. Cu modelul C/C++ ca referință, erorile ar trebui să fie ușor detectabile, deoarece fiecare nou modul este substituit în programul asamblor original produs prin compilare.

Procesul descris mai sus poate fi lent, laborios pentru un codor de vorbire complex. Calea de a-l face mai rapid este de a utiliza o bibliotecă comună de module de codare pentru vorbire. Dacă este adoptată această abordare bazată pe biblioteci, scrierea codorului se va face pe cât posibil prin folosirea modulelor luate din bibliotecă. Pentru aceste module vor exista deja rutine foarte eficiente în limbaj de asamblare. Problema reprezentării variabilelor în aceste module este, de asemenea, deja rezolvată. Ceea ce mai rămâne este de a scrie rutinele în asamblor pentru părțile care nu există încă. Programatori experimentați urmează adesea această cale. Ei refolosesc cât de mult posibil părțile de program existente pentru a-și mări productivitatea.

Se pot cumpăra unele pachete software pentru cipuri DSP cu virgulă fixă, care intenționează să direcționeze întregul proces către o abordare pe baza acestor biblioteci. Pe un procesor gazdă, codorul sau un alt algoritm pot fi construite dintr-un set de funcții din bibliotecă. Simularea inițială trebuie construită în întregime din aceste module, pentru

a câştiga maximul de eficienţă. Două astfel de pachete software sunt Signal Processing WorkSystem® (SPW®) de la Comdisco Systems Inc. şi COSSAP de la CADIS. Aceste pachete nu sunt disponibile pentru toate cipurile DSP. De regulă ele merg doar pentru unul sau două cipuri ale producătorilor. Creatorii algoritmilor pentru sisteme de procesare a semnalelor pot să folosească o abordare a schemei bloc cu aceste sisteme şi să termine cu programe DSP asamblate eficient, cu condiţia ca toate elementele algoritmului să folosească funcţii din bibliotecă.

4.2. *Implementarea cu virgulă mobilă*

Aşa cum s-a văzut din capitolele precedente, implementarea cu virgulă mobilă evită problema determinării reprezentării individuale a diferitelor variabile. Avantajul principal al virgulei mobile este evitarea acestei etape, ceea ce face implementarea cu virgulă mobilă mult mai puţin dificilă decât cu virgulă fixă.

Compilatoare C există pentru mai toate procesoarele cu virgulă mobilă, ceea ce face posibil ca simpla compilare a unei simulări în C să ruleze pe un cip DSP cu virgulă mobilă. Ca şi în cazul virgulei fixe, aceasta serveşte ca un bun model pentru implementarea finală folosind asamblări eficiente. Spre deosebire de cazul virgulei fixe, adesea este dificil să se reproducă cu exactitate aritmetica DSP într-o simulare de nivel înalt. Aceasta deoarece puţine procesoare DSP pot să ruleze eficient aritmetici IEEE în virgulă mobilă. Prin urmare, modelul de referinţă este programul C compilat pentru DSP, nu pentru o maşină de uz general.

Încă o dată, compilatoarele C nu sunt teribil de eficiente pentru cipurile DSP, cu virgulă fixă sau mobilă. În parte aceasta se datorează arhitecturii cipurilor DSP, care nu sunt realizate cu un model de memorie, care să se potrivească bine limbajului C. (În schimb, modelul lor de memorie se potriveşte mai bine limbajului Fortran.) Sunt mai multe căi de a creşte randamentul programelor asamblor, printre altele şi aceea de a le rescrie bătrâneşte, de mână. Multe dintre cele mai populare cipuri DSP cu virgulă mobilă, ca seriile Texas Instruments C3x şi C4x, au biblioteci software care pot fi cumpărate. Aceste biblioteci oferă rutine eficiente pentru funcţii folosite frecvent în procesarea semnalelor. Aceste funcţii pot fi introduse în versiunea finală a programului sursă, reducând sarcina de a scrie programul.

Sistemele de operare DSP care pot fi cumpărate, cum este SPOX de la Spectron Microsystems Inc., conţin aceste biblioteci şi compilatoare „optimizate" C. Sistemul de la AT&T pentru cipul lor DSP3210 conţine un set similar de biblioteci de funcţii, care pot fi combinate pentru o aplicaţie anume, pentru a oferi un mediu multi-tasking în cadrul DSP. Aceasta poate permite unui producător să adune rapid o varietate de aplicaţii diferite, toate rulând pe acelaşi cip, simultan sau secvenţial.

4.3. *Implementarea pe procesoare gazdă*

Procesoarele gazdă au devenit numai recent suficient de puternice pentru a face posibilă implementarea pe ele a codoarelor de vorbire. Spre deosebire de o simulare, unde operarea în timp real nu este o necesitate, aici se cere codorului de vorbire să opereze mai repede decât în timp real. Procesorul gazdă are un sistem de operare în timp real şi sarcini multiple de realizat. În consecinţă, numai o parte din numărul total de MIPS ale procesorului sunt disponibile pentru codarea vorbirii. Este important să nu se obstrucţioneze codarea vorbirii de către altă sarcină importantă a procesorului.

Expertizarea necesar a fi făcută aici este într-un fel diferită de cazul dezvoltării unui DSP. Sistemele de operare ale DSP-urilor cu virgulă mobilă sunt relativ simple. Cele ale

calculatoarelor personale de înalt nivel și ale stațiilor de lucru sunt cu mult mai complicate. Pentru dezvoltarea aplicației este necesară cunoașterea în detaliu a sistemului și în ce măsură codorul de vorbire poate fi integrat între celelalte funcții ale sistemului. Dincolo de această expertizare, se cere cunoașterea posibilităților procesorului și cum se pot realiza rutinele care să ruleze eficient pe el. Compilatoare C pentru optimizare există, dar intervenția pentru ajustări de mână ale programelor C și ale algoritmilor pot determina diferențe semnificative de performanțe. Aici este nevoie de un programator foarte talentat în a optimiza algoritmul, așa cum a fost și cazul procesoarelor DSP, dar în acet caz condițiile diferă prin faptul că implemetarea se face pe suporturi diferite, care sunt procesoarele gazdă.

5. Concluzii

În acest articol am revăzut o varietate a tehnicilor de codare a avorbirii, care sunt curent utilizate. Acestea au inclus pe cel mai popular, codorul cu predicție liniară cu analiză-prin-sinteză, alte codoare mai simple în domeniul timp, codoare în domeniul frecvență cum este codorul de subbenzi și codorul cu transformată adaptivă, codoare de vorbire bazate pe model, cum ar fi codorul cu transformată sinusoidală, codorul cu excitație multi-bandă și codorul cu interpolarea formelor de undă. Am revăzut standardele curente de codoare de vorbire de la International Telecommunications Union, standarde regionale celulare și organisme de standardizare pentru alte aplicații. În final, am văzut cum sunt implementate codoarele de vorbire pe DSP cu virgulă fixă, pe DSP cu virgulă mobilă și pe procesoare gazdă. În general, implementarea cu virgulă fixă este cea mai dificilă, deși există unelte care ajută implementarea cu module destul de standardizate, care pot să formeze un codor de vorbire. Cipurile DSP cu virgulă mobilă sunt folosite în primul rând atunci când cipul va deservi mai multe funcții, în general ca un „slave" (comandat) al unui procesor de uz general. Există, de asemenea, unelte software pentru construirea aplicațiilor pornind de la biblioteci standard pentru cipuri cu virgulă mobilă. În sfârșit, procesoarele gazdă de nivel înalt sunt cele mai noi vehicule pentru implementare. Resursele de calcul ale unui procesor gazdă sunt oarecum limitate față de viteza unității aritmetice din cadrul cipurilor DSP. Ca rezultat, numai codoare de vorbire cu algoritmi simpli pot fi implementate astfel încât să ruleze mai repede pe un procesor gazdă decât în timp real.

Referințe bibliografice

În completare la următoarele referințe bibliografice, informații despre un standard particular de codare a vorbirii sunt disponibile de la organismul de standardizare care a creat standardul.

1. H. Dudley, "The Vocoder", *Bell Lab. Rec.* **18** (1939) 122–126; și H. Dudley, "The Vocoder–Electrical re-creation of speech", *J. Soc. Motion Pict. Eng.* **34** (1940) 272–278.
2. J. R. Boddie, "Digital signal processor", *Bell System Tech. J.* **60**, 7 (1981) 1431–1439.
3. J. R. Cavanaugh, R. W. Hatch și J. L. Sullivan, "Models for the subjective effects of loss, noise, and talker echo on telephone connections", *Bell System Tech. J.* **55**, 9 (1976) 1319–1371.
4. AT&T Microelectronics, "Advanced data sheet DSP1627 digital signal processor", ian. 1995.
5. J. Flaganan, "Voices of men and machines", *J. Acoustics Society of America* **51** (1972) 1385.
6. J. L. Flanagan, K. Ishizaka și K. L. Shipley, "Synthesis of speech from a dynamic model of the vocal cords and vocal tract", *Bell System Tech. J.* **54** (1975) 487.

7. D. W. Petr, "32 kb/s ADPCM-DLQ coding for network applications", *Proc. IEEE GLOBECOM*, dec. 1982, pp. A8.3-1–A8.3-5.
8. B. S. Atal şi M. R. Schroeder, "Adaptive predictive coding of speech signals", *Bell System Tech. J.* **49** (1970) 1973–1986.
9. H. Fletcher şi W. A. Munson, "Relation between loudness and masking", *J. Acoustical Society of America* **9**, 1 (1937) 1–10.
10. H. Fletcher, "The mechanism of hearing as revealed through experiment on the masking effect of thermal noise", *Proc. of National Academy of Sciences*, aprilie 1938, pp. 265–274.
11. B. S. Atal şi M. R. Schroeder, "Predictive coding of speech signals and subjective error criteria", *IEEE Trans. ASSP* **ASSP-27** (1979) 247–254.
12. J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria", *IEEE J. Selected Areas Commun.* **6** (1988) 314–323.
13. R. J. Safranek şi J. D. Johnston, "A perceptually tuned sub-band image coder with image dependent quantization and post-quantization data compression", *Proc. ICASSP '89*, mai 1989, pp. 1945-1948.
14. B. S. Atal şi J. R. Remde, "A new model of LPC excitation for producing natural sounding speech at low bit rates", *Proc. ICASSP '82*, aprilie 1982, pp. 614–617.
15. S. Singhal şi B. S. Atal, "Improving performance of multi-puee LPC coders at low bit rates", *Proc. ICASSP '84*, martie 1984, pp. 1.3.1–4.
16. W. B. Kleijn, D. J. Krasinski şi R. H. Ketchum, "Improved speech quality and efficient vector quantization in SELP", *Proc. ICASSP '88*, aprilie 1988, pp. 155–158.
17. P. Kroon şi B. S. Atal, "Pitch predictors with high temporal resolution", *Proc. ICASSP '90*, 1990, pp. 661–664.
18. R. P. Ramachandran şi P. Kabal, "Pitch prediction filters in speech coding", *IEEE Tans ASSP* **ASSP-37** (1989) 467–478.
19. P. Kroon, E. F. Deprettere şi R. J. Sluyter, "Regular-pulse excitation: A novel approach to effective and efficient multi-pulse coding of speech", *IEEE Trans. ASSP* (1986) 1054–1063.
20. B. S. Atal şi M. R. Schroeder, "Stochastic coding of speech signals at very low bit rates", în *Links for the Future, Science, Systems, and Services for Communications*, redactori P. Dewilde şi C. A. May, Elsevier Science Publications, Amsterdam, 1984, pp. 1610–1612.
21. Federal Standard 1016, "Telecommunications: Analog to digital conversion of radio voice by 4,800 bit/second code excited linear prediction (CELP)", publicată de General Services Administration, feb. 1991.
22. I. A. Gerson şi M. A. Jasiuk, "Techniques for improving the performance of CELP-type speech coders", *IEEE JSAC* **10**, 5 (1992) 858–865.
23. J.-P. Adoul, P. Mabilleau, M. Delprat şi S. Morissette, "Fast CELP coding based on algebraic codes", *Proc. ICASSP '87*, aprilie 1987, pp. 1957–1960.
24. J.-H. Chen, R. V. Cox, Y.-C. Lin, N. Jayant şi M. J. Melchner, "A low-delay CELP coder for the CCITT 16 kb/s speech coding standard", *IEEE JSAC* **10**, 5 (1992) 830–849.
25. N. S. Jayant şi V. Ramamoorthy, "Adaptive postfiltering of 16 kb/s ADPCM speech", *Proc. ICASSP '85*, aprilie 1985.
26. Y. Yatsuzuka, S. Iizuka şi T. Yamazaki, "A variable rate coding by APC with maximum likelihood quantization from 4.8 kbit/s to 16 kbit/s", *Proc. ICASSP '86*, aprilie 1986, pp. 56.12.1–4.
27. J.-H. Chen, "Low-bit-rate predictive coding of speech waveforms based on vector quantization", teza de doctorat, University of California, Santa Barbara, mar. 1987.
28. A. Gersho, "Advances in speech and audio compression", *Proc. of IEEE* **82** (1994) 900–918.

29. R. E. Crochiere, S. A. Webber şi J. L. Flanagan, "Digital coding of speech in sub-bands", *Bell System Tech. J.* **55** (1976) 1069–1085.
30. R. E. Crochiere, "On the design of sub-band coders for low bit-rate speech communications", *Bell System Tech. J.* **56** (1977) 747–770.
31. T. A. Ramstad, "Sub-band coder with a simple adaptive bit allocation algorithm", *Proc. ICASSP '82*, aprilie 1982, pp. 203–206.
32. R. Zelinski şi P. Noll, "Adaptive transform coding of speech signals", *IEEE Trans. ASSP* **25** (1977) 299–309.
33. J. M. Tribolet şi R. E. Crochiere, "Frequency domain coding of speech", *IEEE Trans. ASSP* **27** (1979) 512–530.
34. R. E. Crochiere şi J. M. Tribolet, "Frequency domain techniques for speech coding", *JASA* **66** (1979) 1642–1646.
35. R. V. Cox şi R. E. Crochiere, "Real-time simulation of adaptive transform coding", *IEEE Trans. ASSP* **ASSP-29** (1981) 147–154.
36. B. Mazor, D. Venneman şi E. Mintz, "Adaptive sub-bands excited transform (ASET) coding", *Proc. ICASSP '86*, aprilie 1986, pp. 160–163.
37. F. K. Soong, R. V. Cox şi N. S. Jayant, "A high quality sub-band coder with backward adaptive predictor and optimal time-frequency bit assignment", *Proc. ICASSP '86*, aprilie 1986, pp. 2387–2390.
38. R. V. Cox, S. L. Gay, N. Seshadri, Y. Shoham, S. Quackenbush şi N. S. Jayant, "New directions in sub-band coding", *IEEE J. Selected Areas in Commun.* **6** (1988) 391–409.
39. Y. Shoham şi A. Gersho, "Pitch synchronous transform coding of speech at 9.6 kb/s based on vector quantization", *Proc. ICC '84*, vol. 3, 1984, pp. 1179–1182.
40. V. E. Sanchez şi J.-P. Adoul, "Low-delay wideband speech coding using a new frequency domain approach", *Proc. ICASSP '93*, aprilie 1993, pp. II-415–418.
41. R. Lefebvre, R. Salami, C. Laflamme şi J.-P. Adoul, "High quality coding of wideband audio signals using transform coded excitation (TCX)", *Proc. ICASSP '94*, aprilie 1994, pp. I-193–196.
42. C. Laflamme, J.-P. Adoul, R. Salami, S. Morissette şi P. Mabilleau, "16 kbps wideband speech coding technique based on algebraic CELP", *Proc. ICASSP '91*, mai 1991, pp. 177–180.
43. R. Salami, C. Laflamme şi J.-P. Adoul, "Real-time implementation of a 9.6 kbit/s ACELP wideband speech coder", *Proc. GLOBECOM '92*, dec. 1992, pp. 447–451.
44. E. Harborg, J. E. Knudsen, A. Fuldseth şi F. T. Johansen, "A real-time wideband CELP coder for a videophone application", *Proc. ICASSP '94*, aprilie 1994, pp. II-121–124.
45. Y. Shoham şi E. Ordentlich, "Low-delay code-excited linear predictive coding of wideband speech at 32 kbps", *Proc. Int. Conf. Spoken Language Processing (ICSLP '90)*, noi. 1990, pp. 117–120.
46. O. Gottesman şi Y. Shoham, "Real-time implementation of high quality wideband audio LD-CELP coder", *Proc. EUROSPEECH '93*, sept. 1993, pp. 1115–1118.
47. R. J. McAulay şi T. F. Quatieri, "Speech analysis/synthesis based on a sinusoidal representation", *IEEE Trans. ASSP* **ASSP-34** (1986) 744–754.
48. R. J. McAulay şi T. F. Quatieri, "Low-rate speech coding based on the sinusoidal model", în *Advances in Acoustics and Speech Processing*, redactori M. Sondhi and S. Furui, Marcel Dekker, New York, 1992, pp. 165–207.
49. D. W. Griffin şi J. S. Lim, "Multiband excitation vocoder", *IEEE Trans. ASSP* **36**, 8 (1998) 1223–1235.
50. J. C. Hardwick şi J. S. Lim, "The application of the IMBE speech coder to mobile communications", *Proc. ICASSP '91*, mai 1991, pp. 249–252.

51. W. B. Kleijn şi J. Haagen, "A general waveform interpolation structure for speech coding", în *Signal Processing VII*, redactori M. J. J. Holt, P. M. Grant şi W. A. Sandham, Kluwer Academic Publishers, 1994, pp. 1665–1668.
52. W. B. Kleijn şi J. Haagen, "Transformation and decomposition of the speech signal for coding", *IEEE Signal Processing Lett.* **1** (1994) 136–138.
53. S. Dimolitsas, private communication.
54. T. Tremain, private communication.
55. N. S. Jayant şi P. Noll, *Digital Coding of Waveforms*, Prentice Hall, 1984, pp. 232–233.
56. W. R. Daumer, X. Maitre, P. Mermelstein şi I. Tokizawa, "Overview of the 32 kb/s ADPCM algorithm", *Proc. IEEE GLOBECOM '84*, nov. 1984, pp. 774–777.
57. M. Taka, R. Maruta şi A. Le Guyader, "Synchronous tandem algorithm for 32 kb/s ADPCM", *Proc. IEEE GLOBECOM '84*, nov. 1984, pp. 791—795.
58. M. Taka şi X. Maitre, "CCITT standardizing activities on speech coding", *Proc. ICASSP '86*, aprilie 1986, pp. 817–820.
59. J.-H. Chen şi R. V. Cox, "The creation and evolution of 16 kbit/s LD-CELP: From concept to standard", *Speech Communications* **12** (1993) 103–112.
60. F. T. Johansen, "A non bit-exact approach for implementation verification of the CCITT LD-CELP speech coder", *Speech Communications* **12** (1993) 157–170.
61. C. R. South, J. Rugelbak, P. Usai, N. Kitawaki, H. Irii, J. Rosenberger, J. R. Cavanaugh, C. A. Adesanya, D. Pascal, N. Gleiss şi G. J. Barnes, "Subjective performance assessment of CCITT's 16 kbit/s speech coding algorithm", *Speech Communications* (1993) 113–134.
62. CCITT SG XV, "Terms of reference for the 8 kb/s speech coder", COM XV-R 73-E, ian. 1992, pp. 44–54.
63. ITU-T SG15, "Report of working party 1/15 (audiovisual)", COM 15-R 16-E, 1994, pp. 42–44.
64. "Provisional terms of reference for a future 4 kbit/s speech coding algorithm", ITU-T COM 15-R 19-E, iunie 1994, pp. 47–54.
65. "Provisional terms of reference for wideband (7 kHz) speech coding", ITU-T COM 15-R 19-E, iunie 1994, pp. 38–45.
66. P. Vary, K. Hellwig, R. Hofmann, R. J. Sluyter, C. Galand şi M. Russo, "Speech codec for the European mobile radio system", *Proc. ICASSP '88*, aprilie 1988, pp. 227–230.
67. I. A. Gerson şi M. A. Jasiuk, "Vector sum excited linear prediction (VSELP)", în *Advances in Speech Coding*, redactori B. S. Atal, V. Cuperman şi A. Gersho, Kluwer Academic Publishers, 1991, pp. 69–79.
68. TIA/EIA Interim Standard 85, "Recommended minimum performance standards for full rate speech codes", mai 1992.
69. A. DeJaco, W. Gardner, P. Jacobs şi C. Lee, "QCELP: The North American CDMA digital cellular variable rate speech coding standard", *Proc. IEEE Workshop on Speech Coding for Telecommunications*, oct. 1993, pp. 5–6.
70. TIA/EIA Interim Standard 125, "Recommended minimum performance for digital cellular wideband spread spectrum speech service option 1", aug. 1994.
71. E. De Martino, "Speech quality evaluation of the standards, GSM, TIA and MPT speech coding algorithms for digital cellular systems", *Proc. IEEE Workshop on Speech Coding for Telecommunications*, sept. 1991.
72. T. Miki, "5.6 kb/s PSI-CELP for digital cellular mobile radio", *Proc. First Int. Workshop on Mobile Multimedia Communications*, Tokyo, Japonia, dec. 1993.
73. T. Ohya, H. Suda şi T. Miki, "5.6 kb/s PSI-CELP of the half-rate PDC speech coding standard", *Proc. IEEE Vehicular Technology Conf.*, iunie 1994, pp. 1680–1684.

74. T. Tremain, "The government standard linear predictive coding algorithm: LPC-10", *Speech Technology*, aprilie 1982, pp. 40–49, prelegerea originală.
75. T. E. Tremain, D. P. Kemp. J. S. Collura şi M. A. Kohler, "Evaluation of low rate speech coders for HF", *Proc. ICASSP '93*, aprilie 1993, pp. II-163–166.
76. B. Nouy, P. de la Noue şi G. Goudezeune, "NATO Stanag 4479, a standard for an 800 bps vocoder and redundancy protection in HF-ECCM system", *Proc. ICASSP '95*, mai 1995, pp. 480–483.
77. J. P. Campbell, V. C. Welch şi T. E. Tremain, "The new 4800 bps voice coding standard", *Proc. Military Speech Tech. '89*, nov. 1989, pp. 64–70.
78. M. Kohler şi L. Supplee, "Progress towards a new government standard 2400 b/s voice coder", *Proc ICASSP '95*, mai 1995, pp. 488–491.
79. W. B. Kleijn şi J. Haagen, "A speech coder based on decomposition of characteristic waveforms", *Proc. ICASSP '95*, mai 1995, pp. 508–511.
80. R. V. Fox şi P. Kroon, "Low bit rate speech coders for multimedia communication", *IEEE Communications Magazine*, dec. 1996, pp. 34–40.
81. A. McCree, K. Truong, E. George, T. Barnwell şi V. Viswanathan, "A 2.4 kbit/s MELP coder candidate for the new US Federal Standard", *Proc. ICASSP '96*, mai 1996, vol. 1, pp. 200–203.

Articol din „International Journal of High Electronics and Systems", Vol. 8, No. 1 (1997) 69-118
© World Scientific Publishing Company

CODAREA AUDIO ISO/MPEG

PETER NOLL
Technische Universität Berlin
Einsteinufer 25, D-1000, Berlin 10, Germany

DAVIS PAN
Digital Equipment Corporation
Cambridge Research Lab
One Kendall Square, Bldg. 700
Cambridge, MA 02139, USA

ISO / MPEG (The Moving Pictures Expert Group din cadrul International Organisation of Standardization) a dezvoltat și dezvoltă în prezent o serie de standarde audio-vizuale. Standardul de codare Audio MPEG Faza 1 este primul standard internațional în domeniul compresiei audio digitale de înaltă calitate și a fost aplicat în mai multe domenii, atât pentru aplicații profesionale cât și de consum. Aplicațiile tipice din domeniul audio digital sunt producțiile audio, distribuția și schimburile de programe, transmisia digitală radio, stocarea digitală și variate aplicații multimedia. Acest articol va descrie cu unele detalii principalele caracteristici ale codoarelor MPEG Faza 1.

Ca un pas logic înainte în domeniul audio digital, standardul Audio MPEG Faza 2 este în curs de standardizare, pentru a oferi o mai bună redare stereofonică pentru aplicațiile exclusiv audio, incluzând teleconferințele, cât și pentru sisteme de televiziune îmbunătățite. Stadiul acestui proces de standardizare va fi prezentat pe scurt.

1 Introducere

Asistăm la o creștere a interesului pentru compresia de înaltă calitate a vorbirii de bandă telefonică, a vorbirii de bandă largă și a semnalelor audio de bandă largă.[1-5] Aceste clase de semnale diferă nu numai prin lărgimea lor de bandă și prin plaja lor dinamică, dar și prin așteptările pe care le are ascultătorul asupra calității oferite. Scopurile curente în *codarea vorbirii de bandă telefonică*, cu banda ei de la 300 la 3400 Hz, se concentrează asupra unei calități transparente la frecvențe de bit pornind de la 8 kbps (corespunzând unui bit pe eșantion) în jos. Prin transparență înțelegem că la o ascultare subiectivă a semnalului decodat, nu se observă nici o deosebire față de codorul PCM pe 64 kbps, azi cel mai utilizat standard de codare a vorbirii, ITU-T G.711. Benzi mai largi conduc la îmbunătățiri subiective majore a calității vorbirii reprezentate. O bandă de la 50 la 7000 Hz îmbunătățește inteligibilitatea și naturalețea vorbirii și adaugă o senzație de transparență a comunicării. Aplicații de relevanță mare sunt telefonia „în difuzoare", sistemele de videoconferințe, ISDN și utilizarea canalelor de comentariu pentru transmisiunile radiodifuzate. Activitățile curente în codarea vorbirii de bandă largă se concentrează pe codarea la 16 kHz și mai jos, având ca referință standardul ITU-T G.722 pe 64 kHz (vezi capitolul 5.2).

Discul compact (CD-ul) este astăzi de facto *standardul reprezentativ audio*. El stochează semnale audio cu bandă de la 10 până la peste 20000 Hz, într-un format PCM pe 16 biți, cu o rată de bit rezultantă de 1411,2 kbps pentru stereofonie (vezi capitolul 2.2.). Dacă semnalele trebuie transmise prin canale de capacitate limitată sau trebuie stocate pe medii de capacitate limitată, este obligatoriu să se folosească rate de bit mai mici. Primele propuneri de a reduce ratele PCM au urmat pe cele pentru codarea vorbirii. Oricum, diferențele între vorbire și audio în general, cuprind multe capitole, întrucât codarea audio implică frecvențe de eșantionare mai mari, rezoluția amplitudinilor și plaja dinamică mai bune, variații mari de densitate spectrală de putere, diferențe în ceea ce privește percepția umană și pretenții mult mai mari din partea ascultătorului în privința calității. Codarea vorbirii poate fi atât de eficientă datorită faptului că semnalul de vorbire are la bază un model[NT7] de producere vocală, în timp ce pentru audio în general, nu se poate spune nimic special. Spre deosebire de vorbire, aici trebuie să rezolvăm problemele de stereofonie și de prezentări pe canale multiple.

Aplicațiile tipice pentru spațiul digital audio sunt domeniile producției audio, distribuția și schimbul de programe, al transmisiunilor radiodifuzate cu sunet digital (DSB = *Digital Sound Broadcasting*)[a], al stocării digitale (arhive, studiouri, electronică de consum). Audio digital este, de asemenea, folositor pentru comunicații interpersonale, cum ar fi videoconferințele și aplicațiile multimedia și pentru sistemele de televiziune cu calitate îmbunătățită.

Am văzut progrese rapide în codarea semnalelor vorbirii și audio. Predicția liniară, codarea de subbenzi, codarea prin transformări, ca și diferitele varietăți de cuantizare vectorială și tehnicile de codare, cu îmbunătățirea entropiei, au fost folosite pentru a proiecta algoritmi eficienți de codare, care să ofere mult mai multă compresie decât se putea crede cu numai câțiva ani în urmă. În cazul codării audio, rezultate recente au arătat că o calitate excelentă a codării poate fi obținută sau va fi curând obținută cu rate de bit de numai 2 biți pe eșantion; așteptările pentru următoarea decadă sunt de a reduce această rată cel puțin la 1 bit / eșantion. Aceste mari reduceri sunt posibile prin folosirea unei tehnici numite *codare perceptuală*, prin care se transmit numai acele detalii ale semnalului care pot fi percepute de ureche. Aceste codoare folosesc forme sofisticate de *modelare adaptivă a zgomotului*, controlate pe criterii psiho-acustice.

ISO / MPEG (*The Moving Pictures Expert Group* din cadrul *International Organization of Standardization*) a contribuit semnificativ la sarcina codării informației audio-vizuale. Recentele standarde ISO / MPEG Audio sunt primele standarde internaționale în domeniul compresiei digitale audio de înaltă calitate. De aceea, aceste standarde vor fi descrise în detaliu într-un capitol următor. Fără îndoială, vor fi dezvoltați și alți algoritmi nestandardizați, pentru a servi diferite scopuri precise; în capitolul 5 vom descrie pe scurt câțiva algoritmi de codare diferiți de MPEG.

2. Transmisia, stocarea și radiodifuzarea semnalelor audio

Deși canalele și rețelele de capacitate mare au devenit ușor accesibile, codarea la rate de bit joase a semnalelor audio și-a păstrat importanța. *Principalele motivații pentru codarea la rate de bit reduse* sunt minimizarea costurilor de transmisie sau realizarea

[NT7] Studiat și cunoscut.

[a] Termenul european este DAB = Digital Audio Broadcasting.

stocării eficiente și cerința de a se transmite pe canale de capacitate limitată, cum ar fi canalele radio mobile. În plus, se poate să avem nevoie să distribuim capacitatea unui canal pentru diferite servicii cum ar fi voce, audio, date, grafică și imagini, în servicii integrate de rețea și să folosim codări cu viteze variabile în rețele orientate pe pachete. *Cerințele de bază* în proiectarea codoarelor cu rate de bit reduse sunt în primul rând, menținerea unei înalte calități a semnalului audio refăcut, cu un răspuns bun la variații ale spectrului și ale nivelurilor. În al doilea rând, se cere o imunitate la erori de bit aleatoare în canale zgomotoase și la pierderi de pachete. În al treilea rând, complexitatea redusă și consumul de putere sunt elemente de înaltă relevanță pentru un codor. Un exemplu este complexitatea *decodoarelor* audio, care ar trebui să fie suficient de scăzută pentru a susține soluții ieftine, deoarece aplicațiile de radiodifuzare și reproducere joacă un rol dominant în domeniul audio de bandă largă. Alte *cerințe legate de rețele* sunt întârzieri reduse la codare / decodare, posibilitatea de tandemizare a codecurilor (codec = codor / decodor), posibilitatea de transcodare și degradarea minimă a calității în condițiile creșterii erorilor de bit în aplicațiile radio mobile și radiodifuzate.[6] Toți acești factori aparent contradictorii trebuie considerați cu atenție pentru selectarea unui algoritm de codare audio pentru o anumită aplicație.

2.1. *Canale de transmisie*

Livrarea semnalelor digitale de vorbire și audio este posibilă prin variate canale de transmisie digitală terestră sau bazate pe satelit și sisteme de transmisie cum sunt liniile închiriate, legăturile pentru schimbul de programe, rețelele radio celulare mobile, rețelele TV prin cablu etc. De exemplu, în rețeaua ISDN (*Integrated Service Data Network*) clienții au acces la un număr de canale de telecomunicații (dial-up) relativ ieftine. Tabelul 1 arată câteva canale oferite de ISDN.

Tabelul 1. Canale ISDN de interes pentru codarea audio

Canalul B	64 kbps	B (de la „bearer" = minim)
Canalul D	16 kbps	D (de la date)
Canalul H0	384 kbps	6 x B
Canalul H11	1536 kbps	(= 4 x H0); (S.U.A)
Canalul H12	1920 kbps	(= 5 x H0) ; (Europa)

Interfețele cu rată de bază ISDN constau din două canale B și un canal D de 16 kbps, ultimul suportând semnalizări dar putând să transporte și informații ale utilizatorului. Din aceste cifre este clar că ISDN oferă canale folositoare pentru distribuția practică a semnalelor audio stereofonice și multi-canal. Similar, IVDLAN (*Integrated Voice / Data Local Area Network*; vezi și IEEE 802.9) oferă un număr de canale full-duplex (comunicație în două sensuri) izocrone digitale, (canalele B, C și D) și, în plus, oferă un serviciu tip pachet de bandă largă (canalul P).

Vor fi necesare noi căi pentru viitoarele servicii bidirecționale și servicii „la cerere" (*on demand*) și, în final, ele vor putea fi asigurate pe liniile telefonice. Câtă vreme informația audio însoțește serviciile video digitale, cum sunt serviciile de divertisment și serviciile vizuale interactive, contribuția audio poate fi ușor integrată în fluxul de biți.

2.2. *Mediile de stocare*

Pe un CD sau pe o casetă audio digitală (*DAT = Digital Audio Tape*), semnalele cu o bandă între 10 și 20000Hz sunt stocate cu o rezoluție de 16 biți pe eșantion.[b] La o rată de eșantionare de 44,1 kHz, stereo, rezultă o rată de bit netă de $44,1 \times 2 \times 16 = 1411,2$ kbps. CD-ul necesită un surplus semnificativ de biți pentru o codare liniară a șirurilor uniforme, de lungime limitată (*runlength-limited line code*) care transpune informația din 8 biți în 14 biți și pentru sistemul de sincronizare și de corecție a erorilor, rezultând o reprezentare de 49 de biți pentru fiecare eșantion audio de 16 biți. De aici, rata stereo de bit totală este $1,41 \times 49/16 = 4,32$ Mb / eșantion. Tabelul 2 compară ratele de bit ale CD-urilor și ale benzilor audio digitale.

În domeniul reducerii ratelor de bit, activitățile ISO / MPEG s-au îndreptat inițial către codarea informației audio-vizuale pentru mediile de stocare digitală, cum ar fi discurile magneto-optice (MOD = *Magneto-Optical Disks*), benzile digitale audio (DAT), CD-urile read-only și cele interactive etc.[8] Un prim produs de consum, caseta digitală compactă (DCC = *Digital Compact Cassette*) realizată de Philips, folosește în esență Nivelul 1 al codorului ISO/MPEG.

Un al doilea exemplu de stocare digitală cu compresie audio este *MiniDiscul* (MD) optic sau magneto-optic Sony, de 64 mm, care folosește algoritmul de compresie cu transformare adaptivă (ATRAC = *Adaptive Transform Acoustic Coding*)[9,10] care operează la 292 kbps pentru stereo (vezi tabelul 2). Acest algoritm va fi descris pe scurt într-un capitol următor. Un al treilea exemplu sunt sistemele de redare audio solid-state (de exemplu, pentru anunțuri) cu date comprimate stocate în memorii EPROM, sau cartele inteligente (*smart card*). Un prototip recent de sistem este *Silicon Audio Player* al firmei NEC, care utilizează un decodor MPEG-1 pe un singur cip și oferă 24 de minute stereo la rata de bit recomandată de 192 kbps.[11] Aplicații viitoare vor include servicii de multimedia interactivă și de schimburi mediatice bazate pe diverse carduri de memorie.

Tabelul 2. Rate de bit pentru diverse dispozitive de stocare. (Semnalele stereofonice, eșantionate la 44,1 kHz; de notat că sistemele DAT suportă și rate de eșantionare de 32 și 48 kHz.)

Dispozitiv de stocare	Format	Rata audio	Surplus	Rata totală
Disc Compact (CD)	PCM	1,41 Mbps	2,91 Mbps	4,32 Mbps
Bandă audio digitală (DAT)	PCM	1,41 Mbps	1,67 Mbps	3,08 Mbps
Casetă digitală compactă (DCC)	PASC	384 kbps	384 kbps	768 kbps
MiniDisc (MD)	ATRAC	292 kbps	718 kbps	1,01 Mbps

2.3. *Radiodifuzarea sunetului digital (DSB = Digital Sound Broadcasting)*

Radiodifuzarea digitală terestră sau bazată pe satelit este o sarcină complexă, în particular dacă ascultătorul folosește receptoare mobile sau portabile. Interferențele multiple între căi și atenuările selective sunt cele mai mari impedimente la care trebuie

[b] Algoritmi recenți de codare, inclusiv MPEG Audio, operează cu o plajă dinamică echivalentă cu o rezoluție de 20 de biți pe eșantion sau chiar mai mare. În producția audio sunt în uz rezoluții PCM de până la 24 biți pe eșantion. Sistemele cu 16 biți pe eșantion au un zgomot de fond audibil dacă sunt comparate cu, să zicem, sistemele cu 18 sau 20 de biți pe eșantion.[7]

să ne aşteptăm. În plus, o reţea de canale radiodifuzate poate include secţiuni cu pretenţii diferite de calitate, de la calitatea specifică producţiilor audio până la cea a comentariilor din transmisii. Calitatea specifică unei producţii audio trebuie luată în considerare atunci când au loc editări, tăieri, postprocesări etc. Legături suplimentare (care permit schimbul de programe) şi legăturile de distribuţie (care transmit sunetul la emiţătoare) trebuie să lucreze adecvat pentru a fi capabile să livreze semnale audio de calitate excelentă emiţătoarelor DSB. În plus, legăturile de comentarii trebuie să poată livra voce de calitate excelentă şi programe muzicale la o calitate mai redusă pentru ascultător.

ITU-R a făcut teste extinse pentru a defini codoarele audio pentru radiodifuziunea digitală audio. El recomandă algoritmul de codare MPEG de Nivel II pentru transmisiunile digitale audio la 128 kbps şi pentru suplimentul la 180+12 kbps (acestă rată din urmă este pentru transmisia de date). Pentru legături de comentarii este recomandat codorul MPEG de Nivel III la o rată de 60+4 kbps. Toate ratele sunt pentru canale monofonice cu codare independentă a canalelor stânga şi dreapta. Rata de eşantionare va fi de 48 kHz în toate cazurile.

Sistemul european DAB va urma recomandările ITU-R şi transmisia se va baza pe COFDM (*Coded Orthogonal Frequency Division Multiplex*), care împarte datele de la codor între un număr mare de purtătoare adiacente. Lărgimi de bandă reduse implică durate mari ale simbolurilor. Dacă aceste durate sunt mai lungi decât întârzierea distribuită pe întregul canal şi dacă se asigură întrepătrunderi eficiente între timpi şi frecvenţe şi tehnici de protecţie la erori, sistemul poate să stăpânească problemele asociate cu fadingul multiplu şi selectiv. În plus, datorită robusteţei la propagări pe căi multiple, pot fi folosite pentru serviciile terestre cu *reţele cu o singură frecvenţă*, eficiente spectral, cu transmiţătoare multiple neinterferente.[12] În Statele Unite, un număr de sisteme digitale de radiodifuziune audio au fost propuse pentru testare în 1994 la EIA (*Electronic Industries Association*) în vederea competiţiei pentru standardizare. O decizie poate fi curând aşteptată.

Sistemele avansate de TV digitală asigură livrarea HDTV către marele public prin radiodifuziune terestră şi o varietate de mijloace de transmisiune media alternative şi oferă programe video de înaltă rezoluţie cu o înaltă fidelitate pentru *„audio surround multicanal"*. Rata de bit totală va fi transmisă în banda de frecvenţă a unui canal TV din UHF. Sistemul din Statele Unite, *Grand Alliance HDTV* şi *DVB (Digital Video Broadcast)* din Europa, folosesc amândouă sistemul de compresie Video MPEG Faza 2 şi, pentru transport, un pachet de protocol asemănător cu ATM cu headere / descriptori. Ele diferă prin modul de compresie a semnalului audio. Sistemul Grand Alliance va folosi tehnica de codare cu transformarea AC-3 de la Dolby[13,14], în timp ce sistemul DVB va folosi algoritmul MPEG Audio Faza 2. Ambele sisteme oferă o varietate de reprezentări p/q ale semnalelor cu p canale de sunet pentru faţă şi q canale pentru spate. Exemple importante sunt sunetul surround 3/2, convenţionalul 2/0 stereo şi 1/0 mono (vezi capitolul 7).

3. Mascarea acustică

Mascarea acustică descrie fenomenul prin care un semnal de nivel mic (mascatul) poate deveni neaudibil când un semnal mai puternic (masca) se produce simultan. Acest fenomen poate fi exploatat în codarea vorbirii şi audio printr-o deformare potrivită a zgomotului în procesul de codare. Astfel de deformare a zgomotului poate asigura o

codare de calitate înaltă fără a fi nevoie de a asigura un raport semnal / zgomot foarte bun. Mascarea depinde de distribuția spectrală a măștii și a mascatului și de variațiile lor în timp.

Percepția acustică se bazează pe analiza benzilor critice în urechea internă, unde are loc o transformare din frecvență în poziție de-a lungul membranei bazale (timpanul). Spectrul de putere nu este reprezentat liniar pe o scară a frecvențelor ci pe benzi de frecvență limitate, numite *benzi critice*.[15] Multe efecte psihoacustice sunt compatibile cu o scară a benzilor critice de frecvență, așa numita *„scară Bark"*. De exemplu, atât puterea percepută a unui semnal cât și audibilitatea lui în prezența unui semnal mască sunt diferite pentru semnale din aceeași bandă critică față de semnale care se extind peste mai multe benzi critice. În interiorul unei benzi critice, intensitățile tonurilor individuale sunt sumate de ureche. Sistemul acustic poate fi descris ca un set de filtre trece bandă, constând din filtre puternic întrepătrunse, cu benzi de ordinul a 100 Hz pentru semnale de până la 500 Hz și benzi de până la 5000 Hz pentru semnale de frecvență înaltă. Până la 24000 Hz au fost luate în considerare 26 de benzi critice. Figura 1 compară o scară liniară a frecvențelor (de la 0 la 24000 Hz) cu o scară a benzilor critice. Scara liniară a frecvențelor este împărțită în 32 de subbenzi, fiecare de 750 Hz lărgime; o astfel de divizare în sub-benzi este folosită de codarea MPEG (vezi capitolul 6). Conceptul de bandă critică necesită o remarcă interesantă: benzile critice nu sunt entități fixe ci sunt de fapt definite de evenimentele acustice care ajung la ureche.

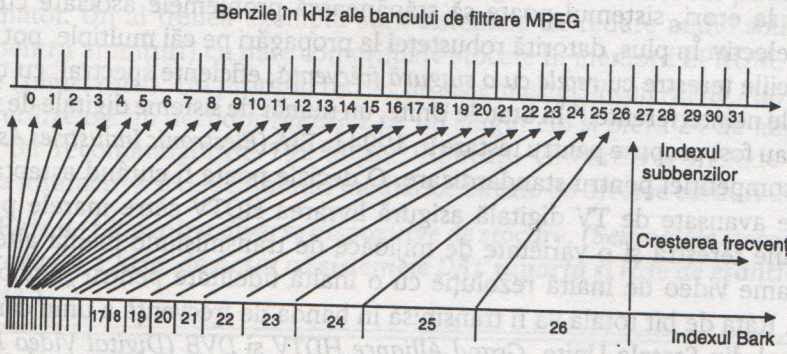


Figura 1. Benzile filtrelor Audio MPEG față de partiția benzilor critice

3.1. *Mascarea simultană*

Mascarea simultană este un fenomen în domeniul frecvențelor: un semnal de nivel mic, fie acesta zgomot, poate fi făcut neaudibil (mascat) prin producerea simultană a unui semnal mai puternic, dacă masca și mascatul sunt suficient de apropiate în frecvență.[15, 16] Fără o mască, un semnal este inaudibil dacă nivelul de presiune sonoră (SPL = *Sound Pressure Level*) este sub *pragul de liniște*, care depinde de frecvență și acoperă o plajă dinamică de aproape 80 dB, așa cum se vede din figura 2 (curba de jos). În această figură, un semnal mască este un zgomot de bandă îngustă (90 Hz) centrat pe frecvența de 1 kHz. În prezența acestui semnal se poate măsura un *prag de mascare*, sub care nici un alt semnal nu va fi auzit. Pragul de mascare depinde de presiunea sonoră (SPL), de frecvența semnalului mască și de caracteristicile semnalelor mască și mascat.

Să luăm de exemplu pragul de mascare pentru un semnal mască cu SPL = 60 dB, ca în figura 2: în jurul lui 1 kHz, nivelul SPL al semnalului mascat poate fi surprinzător de mare, el va fi mascat atâta vreme cât nivelul său este cu numai câțiva decibeli mai mic decât al pragului de mascare! Panta nivelului de prag al mascării este mai abruptă către frecvențele joase, adică frecvențele mari sunt mai ușor de mascat. Schița calitativă din figura 3 dă ceva mai multe detalii despre pragul de mascare: în interiorul unei benzi critice, tonurile de nivel mai mic decât acest prag (zonele înegrite) sunt mascate. Distanța între nivelul semnalului mască și pragul de mascare se numește *raportul semnal / mască (SMR = Signal to Mask Ratio)*. Valoarea lui maximă este la limita stângă a benzii critice (punctul A), valoarea minimă fiind în jurul valorii de 5,5 dB, determinată prin experimente de mascare între zgomot și tonuri. Să presupunem o cuantizare a unui semnal cu *m* biți. Într-o bandă critică, zgomotul de cuantizare nu va fi audibil atâta timp cât raportul semnal / zgomot va fi mai mare decât raportul SMR.

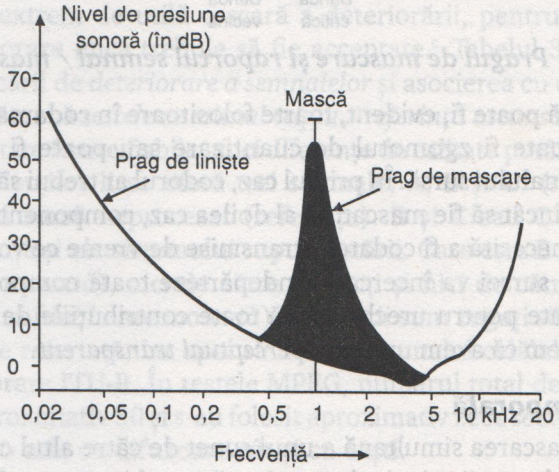


Figura 2. Pragul de liniște (curba joasă) și pragul de mascare. Evenimentele acustice în zona umbrită nu vor fi audibile.

Contribuțiile zgomotului *și* ale semnalului *în afara* benzii critice particulare vor fi și ele mascate dacă nivelul lor de presiune acustică (SPL) este sub pragul de mascare.

Definind SNR(*m*) ca raportul semnal / zgomot care rezultă la cuantizarea cu *m* biți, distorsiunile percepute într-o subbandă dată sunt măsurate prin *raportul zgomot / mască*:

$$\mathrm{NMR}(m) = \mathrm{SMR} - \mathrm{SNR}(m) \text{ (în dB)}. \qquad (3.1)$$

Raportul zgomot / mască NMR(*m*) descrie raportul semnal / mască (SMR) și semnal / zgomot (SNR) la care ne așteptăm în urma cuantizării cu *m* biți. Valoarea NMR este, de asemenea, diferența (în dB) între nivelul zgomotului de cuantizare și nivelul de la care distorsiunile devin audibile într-o subbandă dată.

Am descris mai sus mascarea de către *un singur* semnal mască într-o bandă critică dată. Dacă sursa de semnal constă din mai multe măști simultane, trebuie să luăm în calcul că fiecare dintre ele are propriul ei prag de mascare. O abordare simplă ar fi să presupunem că toate pragurile de mascare individuale pot fi adunate. Pe baza acestei presupuneri se poate ușor calcula un *prag global de mascare*; detalii sunt date în capitolul 6.

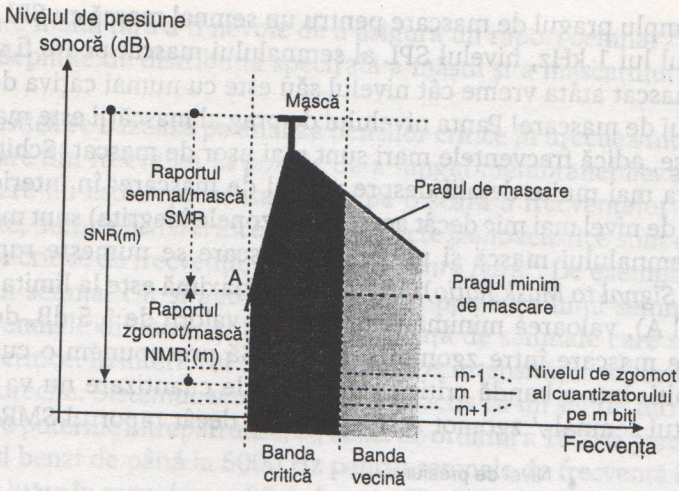


Figura 3. Pragul de mascare și raportul semnal / mască (SMR)

Mascarea acustică poate fi, evident, foarte folositoare în codarea surselor de semnal; semnalul mascat poate fi zgomotul de cuantizare sau poate fi constituit chiar din componente ale semnalului sursă. În primul caz, codorul ar trebui să modeleze zgomotul de cuantizare astfel încât să fie mascat. În al doilea caz, componentele aflate sub pragul de mascare nu mai necesită a fi codate și transmise de vreme ce vor fi oricum mascate. Un codor eficient al sursei va încerca să îndepărteze toate componentele din semnal, care nu sunt relevante pentru ureche. Dacă toate contribuțiile de zgomot pot fi îndepărtate, atunci spunem că avem un *codor perceptual transparent*.

3.2. *Mascare temporală*

În plus față de mascarea simultană a unui sunet de către altul care începe în același timp, mai sunt două domenii principale care joacă un rol important în percepția acustică: *pre-mascarea și post-mascarea*. Efectele de mascare temporală apar înainte și după ce un semnal mască a început, respectiv s-a terminat (figura 4.). Durata cât are loc pre-mascarea este de ordinul câtorva milisecunde, în timp ce post-mascarea este cuprinsă între 50 și 200 ms.

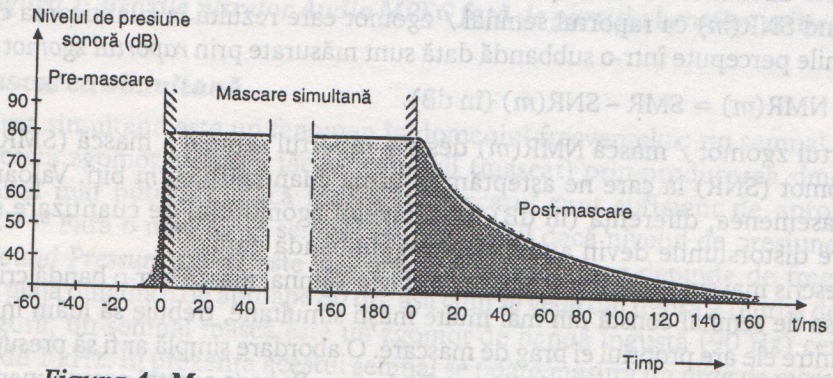


Figura 4. Mascarea temporală.[15] Evenimentele acustice din zonele mai întunecate nu vor fi auzite.

3.3. *Măsurători calitative*

Reprezentarea digitală a formelor de undă analogice cauzează introducerea unui anumit fel de distorsiuni pe care le numim *distorsiuni de codare* sau *zgomot de codare*. Distorsiunile pot fi specificate pe criterii subiective cum sunt scorul mediu de opinie (MOS = *Mean Opinion Score*) ca o măsură a *calității perceptuale*, sau prin scorul mediu de deteriorare ITU-R, ca o măsură a *deteriorării* și, de asemenea, prin criterii obiective simple cum ar fi raportul semnal / zgomot ca o măsură a *similitudinii formelor de undă* ale sursei și ale semnalului refăcut după decodare, sau prin criterii complexe cum ar fi măsurători obiective ale *similitudinii percepției*, care țin cont de particularitățile percepției acustice umane.

Metoda cea mai populară de avizare subiectivă este scorul mediu de opinie unde calitatea codoarelor este clasificată de subiecți, pe o scară a calității cu *N* puncte. Scorul mediu de opinie (valorile MOS) este folosit pentru a evalua codoarele de vorbire. În testele audio, este extrem de utilă o scară a deteriorării, pentru că de obicei numai codoarele cu deteriorare mică trebuie să fie acceptate.[c] Tabelul 3 care urmează arată nivelele ITU-R ale scării de *deteriorare a semnalelor* și asocierea cu un punctaj de 5 note. Testele MPEG au arătat că *testele cu stimul triplu / referință ascunsă / dublu necunoscută* pot conduce la rezultate foarte fiabile, mici diferențe de calitate putând fi ușor detectabile. În aceste teste, subiecților li se oferă trei semnale, A, B și C. (*stimul triplu*). A este totdeauna semnalul sursă neprocesat (referința). B și C sau C și B sunt semnalele sistemului de referință și ale sistemului supus testării. Care este B și care este C nu este cunoscut (*referință ascunsă*), nici de către subiecți și nici de către conducătorul (sau conducătorii) testului (*dublu necunoscută*). Subiecții sunt rugați să decidă care dintre B și C este sistemul de referință, iar apoi să noteze semnalul celălalt, folosind cinci note ale scării de deteriorare ITU-R. În testele MPEG, numărul total de subiecți (ascultători experți) a fost de aproximativ 50 și s-au folosit aproximativ zece secvențe de test, sesiunile fiind realizate stereo atât cu difuzoare cât și cu căști.

Tabelul 3. Scara de deteriorare cu 5 trepte ITU-R

Deteriorarea	Notarea
Imperceptibilă	5
Perceptibilă, dar nesupărătoare	4
Ușor supărătoare	3
Supărătoare	2
Foarte supărătoare	1

Scorul mediu de deteriorare cu 5 note ITU-R a fost folosit de EBU (*European Broadcasting Union*) pentru a defini o calitate țintă pentru codoarele audio: această calitate este numită „indistinctibilă" față de calitatea de CD, dacă pentru 70% dintre materialele folosite în testele subiective, rezultă o încredere de 95% că între CD și sistemul testat calitatea este aceeași.

Măsurătorile obiective încearcă să modeleze felul în care ascultătorul uman percepe semnalele acustice. Avizările obiective ar trebui să se coreleze cu percepția umană pentru

[c] Frecvent termenul MOS este folosit și pentru scorul de deteriorare.

toate distorsiunile care ar putea fi produse în algoritmul de codare și sistemul de transmisie sau memorare. Un număr mare de parametri au fost propuși pentru vorbire și pentru audio. Măsurători bazate pe percepție fac uz de pragurile de mascare derivate din semnalul de intrare, pentru a le compara cu zgomotul real de codare al codorului. Rezultate recente au arătat că măsurătorile bazate pe percepție sunt cu adevărat promițătoare în măsurarea calității. De exemplu, *măsurarea calității audio perceptuale*[17] a fost aplicată pentru semnalele audio în testul ITU-R Digital Audio Broadcasting și a rezultat o corelație de 0,98 cu o deviație standard de 0,17.[18] Un alt set de parametri, incluzând raportul local zgomot / mască și medierea pe toate benzile critice s-a dovedit a fi ușor de implementat și suficient de precis pentru a fi folositor în proiectarea și evaluarea codoarelor.[19] În testele de codare audio ITU-R, corelația a fost de 0,94 cu o deviație standard de 0,27.

4. Codarea sursă

4.1. *Modelarea zgomotului și codarea perceptuală*

Am descris deja dependența percepției acustice umane *de frecvență* și toleranța perceptuală la erori. Acest fapt poate (și trebuie) să influențeze direct proiectarea codoarelor; tehnici de *modelare dinamică a zgomotului* pot crește zgomotul de codare în benzile de frecvență în care el nu este important din punct de vedere al percepției. Modificarea zgomotului trebuie să fie legată de spectrul pe termen scurt al intrării și poate fi făcută în mai multe feluri. Ca exemplu, se poate folosi în scheme predictive *reacția negativă a zgomotului de cuantizare*. Există configurații variate, având în comun o reacție negativă a zgomotului de cuantizare filtrat, la intrarea cuantizatorului.[1] La codarea în domeniul frecvență, căile cele mai ușoare pentru a lua în calcul proprietățile sistemului acustic sunt oferite de o *alocare fixă sau dinamică a biților* (și de aici rezultă contribuția zgomotului de cuantizare) pentru subbenzi sau pentru coeficienții de transformare.

Modelarea zgomotului stă la baza codării eficiente, bazate pe percepție. Figura 5 ilustrează structura unui codor perceptual (etajele tampon / separatoare nu sunt incluse). Procesul de codare este controlat de relația dintre raportul semnal / mască și curba de frecvență, calculată printr-o analiză spectrală a segmentului de sunet care trebuie codat, bazată pe FFT[NT8]. În principiu, se poate folosi orice schemă care poate fi controlată de asemenea informații perceptuale. Dacă rata de bit necesară pentru o mascare completă a distorsiunilor este disponibilă, atunci codorul va fi *transparent*, adică semnalul decodat nu va putea fi distins de semnalul sursă, anume de referința de pe CD. În proiectarea practică a codoarelor perceptuale nu putem merge până la limita mascării sau a distorsiunilor abia perceptibile, atâta timp cât trebuie să luăm în considerare post-procesarea semnalului acustic (de exemplu filtrarea prin egalizatoare) de către utilizatorul final precum și procesele multiple de codare / decodare. Spre exemplu, să considerăm un sistem playback, care accentuează frecvențele înalte și, în particular, unele frecvențe peste frecvența unui anumit ton mască. Zgomotul mascat poate deveni în aceste condiții audibil, atunci când nivelul acestor erori depășește pragul global de mascare (*demascarea*

[NT8] Fast Fourier Transformation = transformata Fourier rapidă

zgomotului). În plus, trebuie să notăm că abilitatea unui anumit codor audio, bazat pe percepție, de a păstra o calitate transparentă, depinde puternic de acuratețea modelului de auz folosit la proiectarea lui. Prin urmare, ca o cerință suplimentară, avem nevoie de o margine de siguranță suficientă în proiectarea practică a codoarelor perceptuale.

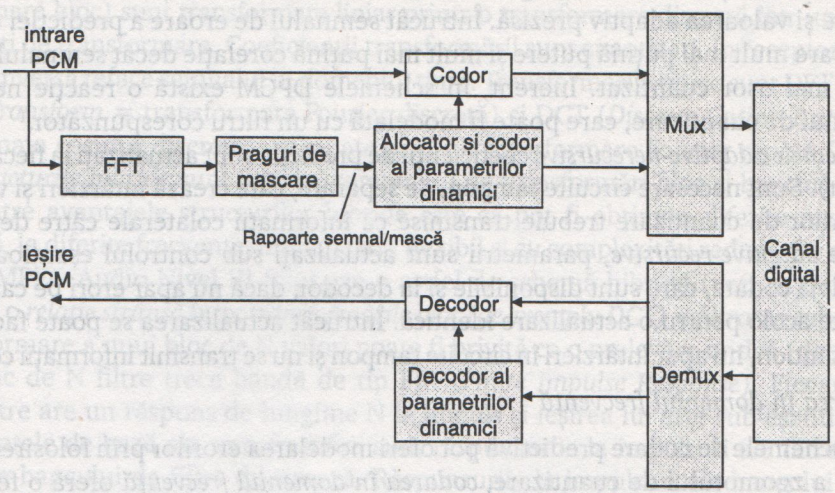


Figura 5. Schema bloc a codării perceptuale

Ar mai trebui notat că recentul standard de codare audio ISO / MPEG este deschis pentru codoare bazate pe modele de localizare psihoacustică mai bune, de vreme ce aceste modele nu sunt elemente normative ale acestui standard (a se vedea capitolul 6.3).

4.2. *Tehnologii cheie*

Codarea digitală la viteze de bit ridicate conservă predominant forma de undă, ceea ce înseamnă că amplitudinea în timp a semnalului decodat o aproximează pe cea a semnalului de intrare. Principiile codării formelor de undă au fost prezentate în detaliu în [1]. La viteze de bit reduse trebuie incluse în proiectarea codorului elemente referitoare la producerea și percepția vorbirii și a semnalelor audio, iar criteriile de eroare trebuie să fie mai degrabă în favoarea unui semnal de ieșire folositor receptorului uman, decât să favorizeze un decodor care să reproducă și să conserve forma de undă de la intrare. În principiu, un algoritm eficient de codare va elimina atât componentele redundante cât și pe cele irelevante din semnalul sursă. Componentele redundante nu trebuie transmise deoarece pot fi refăcute de decodor din datele transmise. Irelevanțele se manifestă ca amplitudini sau frecvențe de o anumită rezoluție nenecesară; în plus, porțiunile din semnalul sursă, care sunt mascate, nu este necesar a fi transmise.

Eliminarea redundanțelor este realizată de codoarele în domeniul timp prin codarea diferențială cu modulația impulsurilor în cod (DPCM = *Differential Pulse Code Modulation*), ori de codoarele în domeniul frecvență prin împărțirea spectrului cu o mare rezoluție și codarea subbenzilor (SBC = *Subband Coding*) sau prin codarea transformatei (TC = *Transform Coding*). Combinații ale acestora sunt de asemenea utilizate. Tehnicile importante de *eliminare a irelevanțelor* sunt cuantizarea adaptivă, cuantizarea vectorială, alocarea dinamică a biților în schemele SBC și TC și modelarea perceptuală a formei spectrale sau temporale a zgomotului (vezi pe pagina următoare).

Modularea diferențială adaptivă a impulsurilor în cod ADPCM (= Adaptive Differential Pulse Code Modulation)

În schemele de ADPCM (modularea diferențială adaptivă a impulsurilor în cod), ceea ce se cuantizează și se transmite digital către decodor este diferența între fiecare eșantion de intrare și valoarea adaptiv prezisă. Întrucât semnalul de eroare a predicției (numit și *reziduu*) are mult mai puțină putere și mult mai puțină corelație decât semnalul sursă, el poate fi mai ușor cuantizat. Inerent, în schemele DPCM există o reacție negativă a zgomotului de cuantizare, care poate fi modelată cu un filtru corespunzător.[1]

În *schemele adaptive-nerecursive* coeficienții de predicție sunt actualizați la fiecare cadru (segment). Sunt necesare circuite tampon, de separare, care crează întârzieri și versiunile parametrilor de cuantizare trebuie transmise ca informații colaterale către decodor. În schemele *adaptive-recursive*, parametrii sunt actualizați sub controlul eșantioanelor de semnal deja codate, care sunt disponibile și la decodor, dacă nu apar erori pe canal și pot fi utilizate acolo pentru o actualizare identică. Întrucât actualizarea se poate face pentru fiecare eșantion, nu apar întârzieri în circuite tampon și nu se transmit informații colaterale.

Codarea în domeniul frecvență

Deși schemele de codare predictivă pot oferi modelarea erorilor prin folosirea reacției negative a zgomotului de cuantizare, *codarea în domeniul frecvență* oferă o formă mai directă de modelare a zgomotului și de suprimare a componentelor de frecvență, care nu este necesar a fi transmise. În codoarele în domeniul frecvență, spectrul sursă este împărțit în benzi de frecvență, fiecare bandă componentă fiind codată separat. De aici rezultă că zgomotul de cuantizare asociat cu o bandă particulară este conținut în acea bandă. Există două scheme de codare, SBC – codarea subbenzilor și ATC – codarea adaptivă a transformatei. Ambele au fost introduse la început în contextul codării vorbirii.[20,21] Diferențierea între aceste două categorii este mai ales de ordin istoric. Ambele folosesc un banc de filtre (ansamblu de filtre) pentru analiză în codor, pentru a produce componente spectrale sub-eșantionate.

În ambele scheme componentele spectrale sunt cuantizate, iar numărul de biți folosiți pentru a coda fiecare componentă spectrală variază: componentele care contribuie mai mult la eroarea medie pătratică sau care sunt subiectiv mai importante, sunt cuantizate mai fin, în timp ce componentele mai puțin importante au mai puțini biți alocați, sau pot să nu fie codate deloc. În funcție de natura semnalului sursă, este necesară o procedură de alocare fixă sau dinamică a biților pentru a obține îmbunătățirea unui codor PCM, bazată pe percepție sau, obiectiv, pe raportul semnal / zgomot. *Alocarea dinamică de biți* a fost introdusă de Zelinski și Noll în 1975 și este o tehnologie cheie în cei mai recenți algoritmi de codare audio.[20,22]

În *codarea subbenzilor*, semnalul sursă este introdus într-un *banc de filtre pentru analiză*, constând în M filtre trece bandă (există și alte structuri, de exemplu arbore cu filtre de două subbenzi cascadate). Ieșirea fiecărui filtru trece bandă este de regulă sub-eșantionată (decimată) cu un factor de M (se prelevează numai un eșantion din M), adică este *eșantionată critic* la o frecvență care este de două ori banda nominală a fiecărui filtru trece bandă. Eșantioanele obținute prin eșantionarea critică a subbenzilor (decimare) sunt cuantizate. În receptor, rata de eșantionare a fiecărei subbenzi este mărită la cea a semnalului sursă prin adăugarea unui număr corespunzător de

eşantioane nule şi semnalele subbenzilor interpolate apar la ieşirile filtrelor trece bandă ale *bancului de filtre de sinteză*. Dacă se pot aplica filtre de refacere perfecte, suma acestor semnale este egală cu semnalul sursă în absenţa cuantizării.

În codarea transformatei, un bloc de N eşantioane de intrare (de aici şi denumirea de transformare bloc) sunt transformate liniar printr-o transformare discretă într-un set de N coeficienţi de transformare. Coeficienţii transformării sunt cuantizaţi. În receptor, o transformare inversă reface semnalul în domeniul timp. Transformările tipice sunt DFT (*Discrete Fourier Transform* = transformata Fourier discretă) şi DCT (*Discrete Cosine Transform* = transformata cosinus discretă), calculate printr-o transformare Fourier rapidă (FFT).

În *structurile hibride* sunt folosite combinaţii de transformări bloc şi bancuri de filtre. Unul dintre avantajele structurilor hibride este că pot fi obţinute diferite rezoluţii de frecvenţă, la diferite frecvenţe, într-un mod flexibil şi cu complexităţi reduse. De exemplu, codorul MPEG Audio Nivel III foloseşte o astfel de schemă hibridă (vezi capitolul 6).

Există o *relaţie strânsă între transformările bloc (de exemplu DCT) şi filtrarea sub-benzilor*. O transformare a unui bloc de N valori poate fi privită ca o prelevare critică (decimare) a unui banc de N filtre trece bandă de tip FIR (*Finite Impulse Response*). Fiecare dintre aceste filtre are un răspuns de lungime N la impuls şi ieşirea lui este sub-eşantionată cu N. Secvenţele de bază ale transformării sunt răspunsurile la impulsuri ale filtrelor trece bandă ale bancului de filtre de sinteză. Răspunsurile la impulsuri ale bancului de filtre de analiză sunt tocmai versiunile inverse în timp ale acestora. Prin urmare, fiecare semnal al unei subbenzi este obţinut prin convoluţia semnalului sursă cu o secvenţă de bază inversată în timp, urmată de sub-eşantionare cu un factor N. Ca un exemplu, să considerăm o transformare DCT[24] ; câteva din secvenţele sale sinusoidale de bază sunt ilustrate în figura 6(a), împreună cu spectrele lor corespunzătoare. De notat că selectivitatea subbenzii este destul de slabă, cauzând o sumă de erori care, oricum, se vor anula una pe alta în transformarea pentru sinteză. Prin folosirea transformării ortogonale închise (*lapped orthogonal transform*) (vezi mai jos) selectivitatea subbenzilor poate fi semnificativ îmbunătăţită, după cum se vede în figura 6(b).

Să mai notăm în trecere că, în mod similar, putem să vedem un banc de filtre ca o transformare cu memorie, în care transformarea unui bloc de semnale poate depinde nu numai de acel bloc, dar şi de cele vecine lui.

Erori în codarea în domeniul frecvenţă

În toate codoarele în domeniul frecvenţă pot fi prezente erori specifice. Întrepătrunderea spectrală a filtrelor de benzi adiacente vor cauza *distorsiuni de întrepătrundere sau de zimţare* (*aliasing distortion*) în domeniul frecvenţă. Dacă la împărţirea pe subbenzi un semnal este divizat în două jumătăţi de bandă suprapuse, se poate aplica o metodă de eliminare a erorilor de întrepătrundere, numită *filtrare în oglindă în cuadratură* (QMF = *Quadrature Mirror Filtering*), care elimină eficient distorsiunile de întrepătrundere.[25] Dacă se doresc mai mult de două benzi, se poate folosi o structură de arbore: fie unul, fie amândouă subbenzile vor alimenta un alt filtru QMF şi se vor diviza în continuare; alte structuri sunt, de asemenea, posibile. În acest fel pot fi realizate diferite lărgimi de bandă şi bancul de filtre poate în principiu să fie adaptat benzilor critice ale semnalului sursă. Un dezavantaj al metodei structurii arborescente este complexitatea mare de calcul şi întârzierea mare. Fiecare divizare adiţională adaugă o întârzire mărită deoarece filtrele operează cu numai o jumătate din rata de eşantionare a etajului precedent.

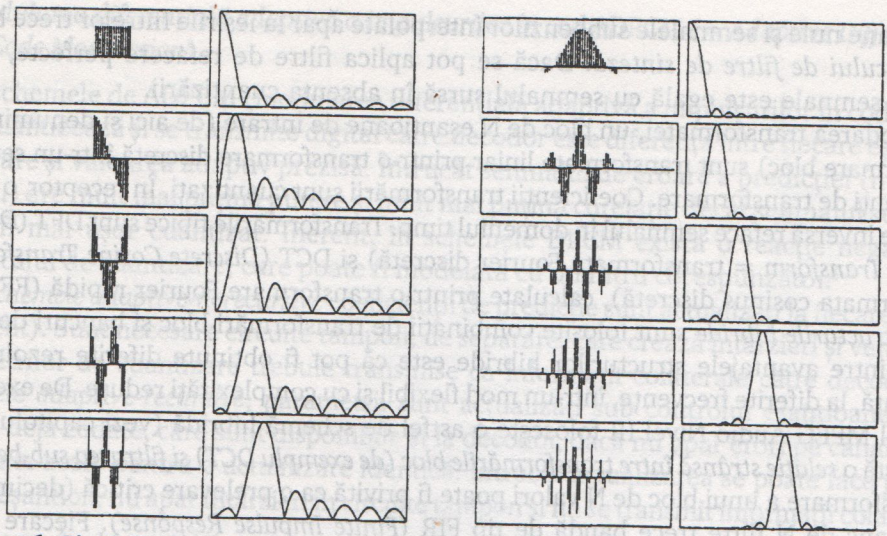


Figura 6. Cinci dintre cele opt secvențe de bază și amplitudinile lor spectrale (0... π)[24] pentru (a) DCT cu 16 puncte (b) Transformarea ortogonală închisă

În plus, rezoluția în frecvență poate fi slabă, deoarece complexitatea de calcul a filtrelor QMF reduce numărul de benzi ce poate fi folosit. Alte categorii de bancuri de filtre de tipul QMF sunt structurile polifază și pseudo-QMF[26]; acestea elimină precis distorsiunile de întrepătrundere ale benzilor adiacente, sunt eficiente din punct de vedere al capacității de calcul de vreme ce se poate utiliza în procesul de filtrare o FFT, au o complexitate moderată și o întârziere redusă. Dezavantajul este că filtrele sunt egal spațiate și de aceea frecvențele benzilor nu corespund corect benzilor critice la frecvențe joase. În codoarele și decodoarele Audio MPEG se folosește un banc de filtre polifază cu 32 de benzi și 512 coeficienți pentru fiecare filtru pentru a obține o atenuare în banda de oprire de cca. 120 dB (vezi capitolul 6).

Codoarele de transformate au în general lungimea blocurilor *N* mare; corespunzător, ele au o rezoluție bună în frecvență. Lungimi de valoare *N* ale funcțiilor de bază corespund unor filtre al căror răspuns la impulsuri se termină abrupt, cauzând așa numitele *efecte la granița între blocuri* (*block boundary effects*). Aceste efecte se pot reduce printr-o mică întrepătrundere a blocurilor procesate succesiv, cu prețul unei creșteri a ratei de bit. Ca alternativă, o DCT (tranformata cosinus discretă) modificată (MDCT) poate să elimine esențialmente efectele la graniță ale blocurilor prin folosirea unei tehnici numite *eliminarea în domeniul timp a întrepătrunderilor* (*TDAC = Time Domain Aliasing Cancellation*).[27] MDCT folosește de obicei o întrepătrundere de 50% a ferestrelor între două blocuri de transformate adiacente. Aceste transformări implementează eficient un banc de filtre bazate pe transformarea unor eșantioane prelevate critic. Ele nu au efecte la graniță între blocuri, au un câștig de codare a transformatei mai mare decât DCT și funcțiile lor de bază corespund unui răspuns mai bun al filtrelor trece-bandă. În ciuda unor erori severe de întrepătrundere în domeniul timp introduse de sub-eșantionare, este posibilă refacerea perfectă a semnalului, deoarece termenii de erori de întrepătrundere se anulează reciproc în etajul de sinteză, atâta timp cât erorile de cuantizare în domeniul transformării pot fi neglijate.

O parte crucială a codării vorbirii şi a semnalelor audio în domeniul frecvenţă este apariţia *pre-ecoului.* Să considerăm cazul în care o perioadă de linişte este urmată de sunet percutant, cum ar fi cele provenind de la castaniete sau triunghiuri muzicale, în cadrul aceluiaşi bloc al unei transformări. Un astfel de „atac" ar cauza erori instantanee de cuantizare comparativ mari. În ATC, transformarea inversă a decodorului va distribui aceste erori asupra întregului bloc; similar, în SBC, ieşirea va fi afectată de aceste erori în concordanţă cu lungimea răspunsului filtrului trece bandă la impuls. În ambele configuraţii apar pre-ecouri şi pot fi auzite distinct, mai ales la rate de bit mici cu erori comparativ mari. Figura 7 demonstrează efectul la codarea cu transformare: dacă dimensiunea blocului este $N=1024$ (figura 7(b)) pre-ecoul este clar (vizibil şi) audibil, în timp ce o dimensiune a blocului de 256 va reduce aceste efecte, deoarece ele sunt limitate la blocul unde apare atacul semnalului şi erorile corespunzătoare de cuantizare (figura 7(c)). În plus, efectul în domeniul timp al mascării inverse poate masca pre-ecouri audibile dacă sunt de lungime mică (de ordinul câtorva milisecunde).

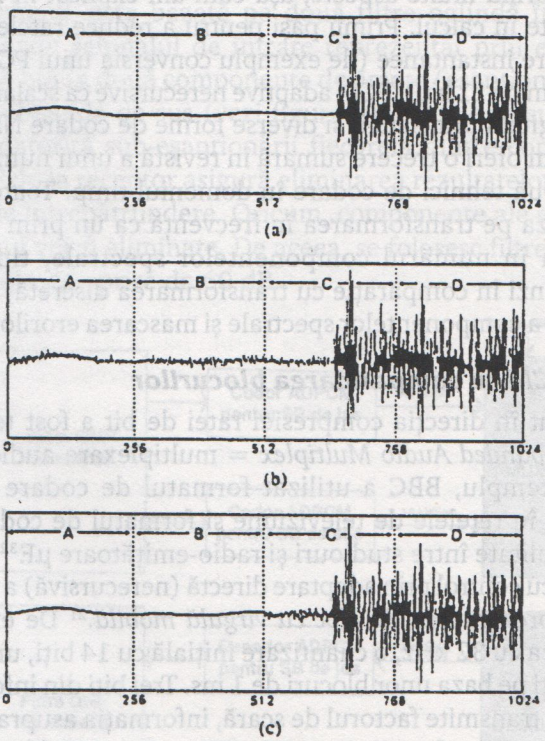


Figura 7. Semnale în codarea transformatei.[29] (a) semnalul-sursă (b) semnalul refăcut cu o dimensiune a blocului N=1024 (c) semnalul refăcut cu o dimensiune a blocului N=256

Deşi audibilitatea pre-ecoului poate fi redusă prin menţinerea dimensiunii mici a blocurilor, eficienţa acestor codoare este diminuată dacă blocurile sunt prea mici, (i) deoarece rezoluţia în frecvenţă este afectată şi (ii) deoarece un procent mai mare din rata de bit va fi necesară transmiterii informaţiilor colaterale. O soluţie a acestei probleme este de a comuta între diferite dimensiuni ale blocurilor de diferite lungimi, de exemplu

N=256 și N=1024 (*comutare adaptivă a ferestrei*).[28,29] Blocurile mici se folosesc numai pentru a controla artefactele (produsele spectrale false) introduse de pre-ecouri în timpul perioadelor nestaționare ale semnalului, altfel codorul se comută pe lungimea mare a blocurilor pentru a păstra câștigul de codare a transformatei cât mai mare posibil. Selectarea dimensiunii blocului trebuie făcută pe baza unei analize pe termen scurt a caracteristicii semnalului audio. În plus față de comutarea adaptivă a ferestrei, audibilitatea pre-ecoului poate fi redusă prin algoritmi de pre- sau postfiltrare, sau prin includerea unui control al pre-ecoului în calcularea globală a pragului de mascare.[30,31]

5. Codarea semnalelor audio

Tabelul 2 a arătat deja principalii parametri ai discului compact (CD), astăzi un cuasistandard în domeniul audio digital. Imediat după apariția codării vorbirii, au apărut propuneri pentru a reduce rata de bit a PCM. Oricum, diferențele între semnalele audio și cele de vorbire cuprind multe aspecte, așa cum am explicat în introducere, și aceste diferențe trebuie luate în calcul. Primii pași pentru a reduce ratele de bit s-au bazat pe tehnici de compandare instantanee (de exemplu conversia unui PCM uniform de 14 biți într-un PCM neuniform de 11 biți), PCM adaptive nerecursive ca scalarea 16/14 în sistemele de radiodifuziune digitală prin satelit și diverse forme de codare NICAM (vezi mai jos).

În continuare, vom oferi o trecere sumară în revistă a unui număr de codoare audio. Vom descrie întâi două tehnici de codare în domeniul timp. Toate exemplele care vor urma apoi se vor baza pe transformarea în frecvență ca un prim pas al codării audio. Diferențele constau în numărul componentelor spectrale, tipul de transformare (împărțirea în subbenzi în comparație cu transformarea discretă) și în strategii pentru cuantizarea eficientă a componentelor spectrale și mascarea erorilor de codare rezultate.

5.1. *PCM și ADPCM cu compandarea blocurilor*

Un pas de început în direcția compresiei ratei de bit a fost tehnica NICAM (*Near Instantaneously Companded Audio Multiplex* = multiplexare audio compandată cvasi-instantaneu). De exemplu, BBC a utilizat formatul de codare NICAM 728 pentru transmisia sunetului în rețelele de televiziune și formatul de codare NICAM 3 pentru legăturile radio de calitate între studiouri și radio-emițătoare μF.[32] Tehnica NICAM este un exemplu de PCM cu control prin adaptare directă (nerecursivă) a pasului de cuantizare și este bazată pe reprezentarea de *bloc cu virgulă mobilă*.[21] De exemplu, NICAM 728 folosește eșantionarea cu 32 kHz, o cuantizare inițială cu 14 biți, urmată de o compresie la un format de 10 biți pe baza unor blocuri de 1 ms. Trei biți din informația suplimentară sunt folosiți pentru a transmite factorul de scară, informația asupra plajei de cuantizare. Rata de bit netă pe un canal audio este $10 \times 32 + 3 = 323$ kbps. Un surplus pentru detecția erorilor și sincronizare mărește rata la 364 kbps pentru un canal mono și la 728 kbps pentru modul stereo (de aici și denumirea NICAM 728). NICAM poate rezolva problema asigurării unei plaje dinamice suficiente pentru codarea audio dar nu și a unei compresii eficiente, deoarece nu exploatează dependențele statistice între eșantioane și nici efectele de mascare perceptuală.

Ca un exemplu de codare adaptivă cu predicție menționăm recentul CD-i (CD-interactiv) care suportă un PCM pe 16 biți la o rată de eșantionare de 44,1 kHz și trei nivele de PCM Adaptivă Diferențială cu 37,8 kHz / 8biți, 37,8kHz / 4 biți și 18,9kHz / 4 biți.

Acest algoritm ADPCM CD-*i* foloseşte un cuantizator adaptiv. Predicţia este de ordin mic, dar predictorul poate fi comutat între diferite seturi, în funcţie de bloc. CD-*i* face uz de discuri de 12 cm care stochează 70 minute de semnale audio şi video în mişcare (full motion), unde partea de video foloseşte un algoritm de codare Video MPEG Faza 1.[d] Rata de bit globală este în jur de 1,4 Mbps, iar calitatea imaginii este comparabilă cu cea a unei înregistrări analogice VHS.

5.2. *Codarea subbenzilor*

În 1996, ITU-T a aprobat un algoritm de *codare a vorbirii de bandă largă* pe 64 kbps (ITU-T G.722) cu banda la 3 dB de 50-7000 Hz.[33] Acest standard este, de asemenea, folositor la codarea semnalelor audio. El foloseşte o tehnică de subbenzi cu două-benzi şi codare adaptivă diferenţială PCM (ADPCM) în fiecare din cele două benzi (figura 8). Rata de bit este 64 kbps, din care opt sau şaisprezece kbps pot fi folosiţi opţional pentru transmisii de date, lăsând 56 sau 48 kbps pentru codarea semnalului de vorbire de bandă largă.

Împărţirea în subbenzi este bazată pe două filtre oglindă în cuadratură (QMF) identice, pentru a divide semnalul de intrare reprezentat prin eşantioane la 16 kHz cuantizate PCM pe 14 biţi în două componente decimate (eşantionate la 8 kHz), numite subbanda de jos şi subbanda de sus (vezi figura 8). Filtrele se suprapun şi apar erori de întrepătrundere datorită sub-eşantionării fiecăreia dintre componente. Bancul de filtre de sinteză QMF de la receptor asigură eliminarea rezultatelor erorilor (produselor spectrale parazite) de întrepătrundere. Oricum, componente ale erorilor de cuantizare din cele două benzi nu vor fi eliminate. De aceea, se folosesc filtre QMF cu 24 de prize, cu o atenuare în banda de oprire de 60 dB.

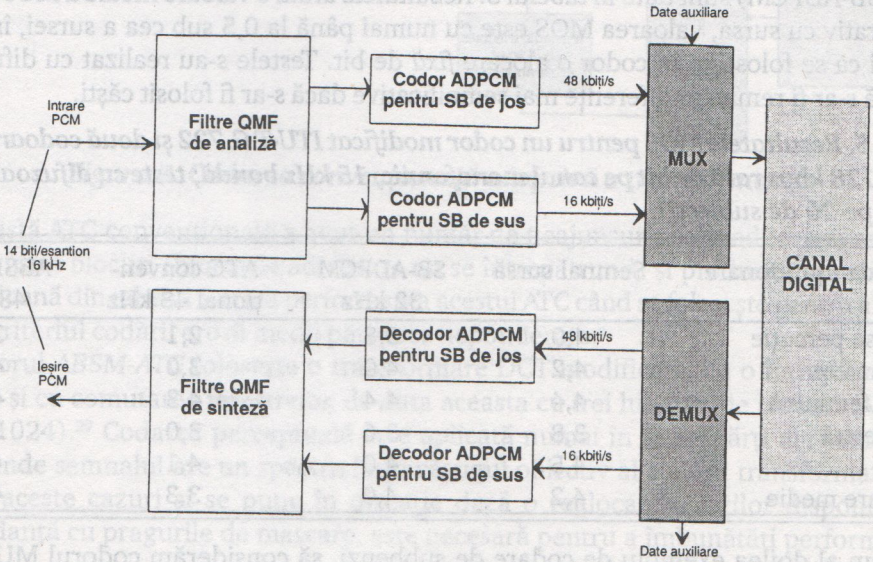


Figura 8. Structura codorului de vorbire de bandă largă ITU-T G.722

[d] Un flux de biţi ISO/MPEG suportă atât semnale numai audio cât şi semnale multiplexate audio-vizuale.

Codarea semnalelor din subbenzi se bazează pe o versiune modificată a codorului de vorbire ADPCM pe 32 kbps ITU-T G.721.[34] Eşantioanele de intrare sunt prezise adaptiv, semnalul eroare de predicţie (sau diferenţa) este cuantizat şi transmis. *Predictorul* este adaptiv recursiv, adică, coeficienţii predictorului sunt actualizaţi pe baza eşantioanelor, sub controlul semnalelor diferenţă deja codate, care sunt disponibile şi la decodor. Predictorul foloseşte o structură cu doi poli şi şase zerouri. *Cuantizatorul* este, de asemenea, adaptiv recursiv şi se poate adapta rapid la schimbările statistice ale semnalului de vorbire. După transmisia erorilor, atât predictorul cât şi cuantizatorul converg (pe termen lung) către valori identice o dată ce nu se mai observă erori de transmisie. Codorul foloseşte o alocare fixă de bit, unde codoarele ADPCM ale subbenzilor de jos şi de sus folosesc eşantioane de 6 biţi / eşantion, respectiv de 2 biţi / eşantion. Teste subiective au arătate că o sursă de vorbire necodată are acelaşi MOS ca şi versiunea sa de 64 kbps, nerezultând diferenţe măsurabile în calitatea subiectivă. Pentru semnalele audio se poate obţine o calitate suficient de bună. Tabelul 4 compară valorile MOS pentru semnale de vorbire şi audio la două rate de bit.

Tabelul 4. Performanţele MOS ale codorului de vorbire de bandă largă ITU-T G.722[33]

Rata de bit (kbps)	Valoarea MOS pentru vorbire	Valoarea MOS pentru audio
48	3,8	3,2
64	4,2	3,9

O versiune modificată a standardului ITU-T G.722 foloseşte o frecvenţă de eşantionare de 32 kHz pentru semnalele audio cu banda de 15 kHz.[29] La o rată de bit rezultată de 128 kbps pe canal monofonic, punctajele MOS ale acestui algoritm de codare adaptiv de subbenzi (SB-ADPCM) sunt date în tabelul 5. Rezultatele arată o valoare medie a MOS de 4,0. Comparativ cu sursa, valoarea MOS este cu numai până la 0,5 sub cea a sursei, în ciuda faptului că se foloseşte în codor o alocare *fixă* de bit. Testele s-au realizat cu difuzoare; poate că s-ar fi remarcat diferenţe mai semnificative dacă s-ar fi folosit căşti.

Tabelul 5. Rezultatele MOS pentru un codor modificat ITU-T G.722 şi două codoare ATC[29] (128 kbps rată de bit pe canal monofonnic, 15 kHz bandă; teste cu difuzoare, bazate pe 20 de subiecţi)

Rată de eşantionare	Semnal sursă	SB-ADPCM 32 kHz	ATC conven-ţional 48 kHz	ABSM-ATC 48 kHz
Cabasa percuţie	4,0	3,8	2,1	3,6
Castaniete	4,2	4,0	3,0	4,0
Voce feminină	4,4	4,4	4,3	4,4
Orchestră	3,8	3,6	3,0	3,5
Pian	4,5	4,0	4,1	4,5
Valoare medie	4,2	4,0	3,3	4,0

Ca un al doilea exemplu de codare de subbenzi, să considerăm codorul MUSICAM (*Masking Pattern Adapted Universal Subband Integrated Coding And Multiplexing*), dezvoltat de CCETT, IRT şi Philips.[35] Acest codor foloseşte o structură de filtru polifază cu filtre cu 512 coeficienţi pentru transformarea intrării digitale audio în 32 de subbenzi egal spaţiate. Cuantizarea este bazată pe o alocare dinamică de bit, care este

controlată de un model perceptual. Deoarece acest concept MUSICAM este baza Nivelurilor I şi II ale codării Audio ISO / MPEG, o vom descrie în detaliu în capitolul 6.

5.3. *Codarea transformatei*

Codarea adaptivă a transformatei (ATC) a fost propusă de Zelinski şi Noll în 1975.[20,22,23] Două dintre principalele caracteristici ale acestei ATC „convenţionale" au fost folosirea transformatei cosinus discrete (DCT) şi introducerea tehnicii de *alocare adaptivă de bit*. DCT transpune un bloc de N eşantioane într-un număr echivalent de coeficienţi de tranformare (vezi figura 9). Din aceşti coeficienţi, se obţine o înfăşurătoare logaritm-spectrală pe termen scurt bazată pe DCT, a cărei aproximare se transmite o dată pe bloc (împreună cu coeficienţii de transformare); această informaţie suplimentară este apoi folosită atât în codor cât şi în decodor, pentru alocarea biţilor.

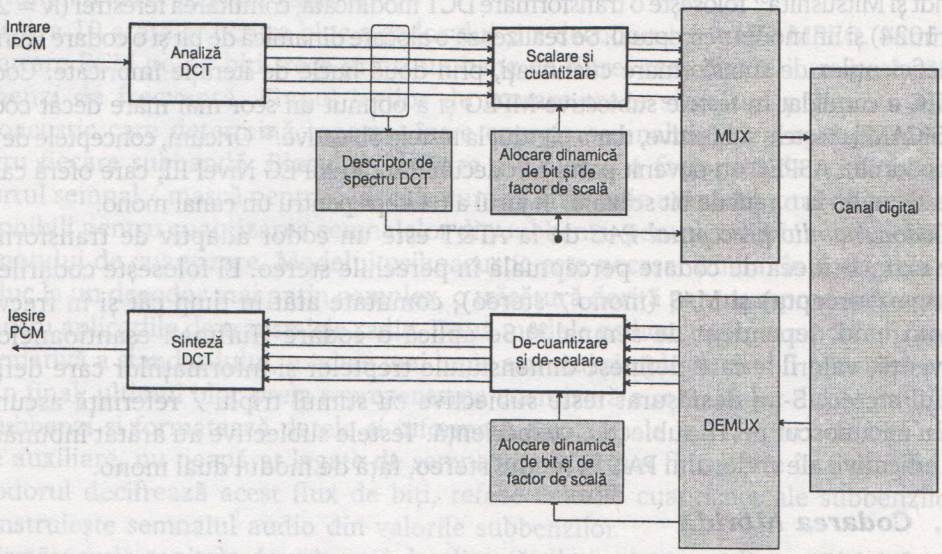


Figura 9. Codarea convenţională adaptivă a transformatei

Această ATC convenţională a avut un număr de neajunsuri: ea producea zgomote la graniţa între blocuri (blocurile adiacente nu se întrepătrund) şi producea pre-ecouri. A treia coloană din tabelul 5 arată performanţa acestui ATC când se foloseşte pentru alocarea biţilor criteriul codării erorii medii pătratice neponderate.[e]

Codorul *ABSM-ATC* foloseşte o transformare DCT modificată, cu o întrepătrundere de 50% şi cu comutarea ferestrelor, de data aceasta cu trei lungimi de blocuri (N=256, 512 şi 1024).[29] Codarea perceptuală este aplicată numai în acele părţi ale semnalului audio unde semnalul are un spectru larg; câştigul obiectiv al codării transformatei este mic în aceste cazuri şi se pune în discuţie dacă o realocare a biţilor disponibili, în concordanţă cu pragurile de mascare, este necesară pentru a îmbunătăţi performanţele

[e] În [20] s-a propus să se îmbunătăţească performanţa prin alocarea dinamică a biţilor disponibili, astfel încât *distorsiunea percepută* să fie minimă. Ca un prim pas, s-a propus un amestec de alocare de bit fixă şi adaptivă, pentru a asigura disponibilitatea biţilor pentru coeficienţii de transformare cărora altfel le-ar fi fost alocaţi zero biţi.

subiective ale codorului. Altfel, compresia ratei de bit se bazează mai mult pe eliminarea redundanțelor decât pe eliminarea părților irelevante.

Codorul *AC-2* de 128 kbps de la Dolby[36] folosește o eliminare a distorsiunilor de întrepătrundere, bazată pe o aplicare alternativă a MDCT și MDST (transformarea sinus discretă modificată) cu 512 puncte. Coeficienții de transformare adiacenți sunt apoi grupați în subbenzi de lărgimi care aproximează benzile critice ale auzului uman. Din coeficienții de transformare din cadrul acestor subbenzi se estimează o înfășurătoare spectrală (anvelopă) logaritmică a blocului curent și aceasta este folosită pentru rutina de alocare dinamică a biților. O versiune modificată a codorului AC-2 a fost evaluată de ITU-R în procesul de standardizare a radiodifuzării audio digitale și s-a dovedit a fi din punct de vedere al performanțelor, aproape de algoritmul de codare MPEG Nivelul II, cu rata lui de bit fixă.

Codorul *ASPEC (Audio Spectro-Perceptual Entropy Coding)* propus de AT&T, FhG, Thomson-Brandt și Mitsushita[37] folosește o transformare DCT modificată, comutarea ferestrei ($N = 256$ și $N = 1024$) și un model perceptual. Se realizează o alocare dinamică de bit și o codare Huffman a coeficienților de transformare cuantizați, prin două bucle de iterație îmbricate. Codorul ASPEC a candidat în testele subiective MPEG și a obținut un scor mai mare decât codorul MUSICAM la testele subiective, dar a pierdut la testele obiective.[53] Oricum, conceptele de bază ale codorului ASPEC au devenit parte a codecului Audio MPEG Nivel III, care oferă calitate ridicată audio la o rată de bit scăzută, în jurul a 64 kbps pentru un canal mono.

Codorul audio perceptual PAC de la AT&T este un codor adaptiv de transformată, care extinde ideea de codare perceptuală în perechile stereo. El folosește codările L/R (stânga / dreapta) și M/S (mono / stereo), comutate atât în timp cât și în frecvență într-un mod dependent de semnal.[38] Se aplică o codare Huffman eșantioanelor de frecvență, valorilor care definesc dimensiunile treptelor și informațiilor care definesc modul stereo. S-au desfășurat teste subiective cu stimul triplu / referință ascunsă / dublu necunoscut pe 18 subiecți cu experiență. Testele subiective au arătat îmbunătățiri semnificative ale codorului PAC în modul stereo, față de modul dual mono.

5.4. *Codarea hibridă*

Codorul *ATRAC (Adaptive Transform Acoustic Coding)* de la Sony a fost dezvoltat pentru sisteme audio digitale portabile, în special pentru discurile optice sau magneto-optice de 64 mm, care au aproximativ o cincime din capacitatea unui CD obișnuit.[9] Codorul utilizează o transformare hibridă a frecvențelor: semnalul este despărțit în trei subbenzi (0-5,5; 5,5-11 și 11-22 kHz) folosind filtre QMF. Se aplică apoi tuturor subbenzilor o transformare MDCT cu 50% întrepătrundere, cu comutare de fereastră (de obicei $N = 512$, dar în cazul apariției unor pre-ecouri N este comutat la 64 în benzile de frecvență înaltă și la $N = 128$ în rest). ATRAC nu specifică un algoritm de alocare de bit. O sugestie este de a folosi o varietate de mixări între alocarea de bit fixă și adaptivă.

Codecul Audio MPEG Nivel III este al doilea exemplu de codare hibridă și va fi descris în capitolul următor.

6. Codecul audio ISO / MPEG

Primul standard audio ISO / MPEG este o realizare a ISO / Moving Pictures Expert Group, care și-a luat sarcina de a dezvolta un standard pentru codarea imaginilor în mișcare asociate cu audio, pentru stocarea digitală pe un mediu până la 1,5 MHz.

Mulţi membri din domeniul telecomunicaţiilor, electronicii de consum, industria calculatoarelor, universităţi şi radiodifuziune au contribuit la realizarea obiectivului ISO / MPEG. Partea normativă a standardului descrie fluxul de biţi şi procesul de *decodare*; procesul de codare trebuie să corespundă fluxului de biţi codat şi este descris în partea informativă a standardului.[8] Un raport interim despre procesul de standardizare şi o primă descriere a standardului final au fost date în [39], respectiv [40].

Standardul audio constă din trei *niveluri* (coduri) de complexitate din ce în ce mai ridicată şi de performanţe subiective mai bune.[8] Standardul prevede rate de eşantionare de 32, de 44,1 şi de 48 kHz şi rate de bit pe canal monofonic între 32 şi 192 kbps. Modurile de operare sunt mono şi stereo, dual (pentru a oferi programe bilingve) şi opţional modul *joint* stereo (stereo multicanal) (în care cele două codoare pentru stânga şi dreapta se susţin unul pe altul prin exploatarea dependenţelor statistice şi irelevanţelor dintre aceste canale).

Figura 10 arată structura bloc a codorului şi a decodorului Audio MPEG. Semnalul de intrare PCM pe 16 biţi trece printr-un banc de filtre, care divide intrarea în 32 de subbenzi de frecvenţă. Fluxul biţilor de intrare trece simultan printr-un model psihoacustic care determină raportul între energia semnalului şi pragul de mascare pentru fiecare subbandă. Blocul de alocare a biţilor şi a factorilor de scală foloseşte raportul semnal / mască pentru a decide cum să distribuie numărul total de biţi de cod disponibili pentru cuantizarea semnalelor din subbenzi, pentru a diminua audibilitatea zgomotului de cuantizare. Modelul psihoacustic este necesar numai în codoarele care conduc la un decodor mai puţin complex, o trăsătură dorită pentru reproducerea audio şi pentru aplicaţiile de transmisie radio. Două modele psihoacustice sunt date în partea informativă a standardului, modele mai bune sau mai simple putând fi folosite în locul lor. În final, ultimul bloc preia reprezentarea cuantizată a semnalelor din eşantioanele de subbenzi şi formatează datele şi informaţiile colaterale într-un flux de biţi codat. Date auxiliare, nu neapărat legate de semnalul audio, pot fi inserate în fluxul de biţi. Decodorul decifrează acest flux de biţi, reface valorile cuantizate ale subbenzilor şi reconstruieşte semnalul audio din valorile subbenzilor.

Următoarele capitole descriu mai detaliat părţile importante din standardul Audio MPEG. Întâi este discutat bancul de filtre polifază folosit de toate nivelele standardului. Capitolul 6.2 va prezenta metodele de implementare eficientă a bancului de filtre. Capitolele 6.3 şi 6.4 se ocupă de cele două modele psihoacustice descrise în standard şi dau un exemplu despre modul în care aceste modele se comportă cu anumite tipuri de semnal audio. Apoi sunt descrise detalii ale diferenţelor care există între cele trei nivele opţionale ale standardului. În final, ultimul capitol se ocupă de algoritmii de alocare de bit şi de zgomot, de codarea stereo multicanal şi aprecieri subiective.

6.1. *Bancul de filtre polifază*

Acest capitol se va ocupa de câteva aspecte ale comportării bancului de filtre de analiză din codorul Audio MPEG. O tratare similară se aplică şi bancului de filtre de sinteză din decodor. Bancul de filtre polifază este componenta cheie comună tuturor nivelurilor compresiei Audio MPEG. Acest banc de filtre cu 512 coeficienţi de filtrare divide zona de frecvenţe a semnalului audio de intrare în 32 de subbenzi de lărgimi de bandă egale. Semnalele de ieşire ale filtrelor trece bandă sunt sub-eşantionate critic (decimate). La o rată de eşantionare de 48 kHz, fiecare subbandă are o lărgime de 750 Hz,

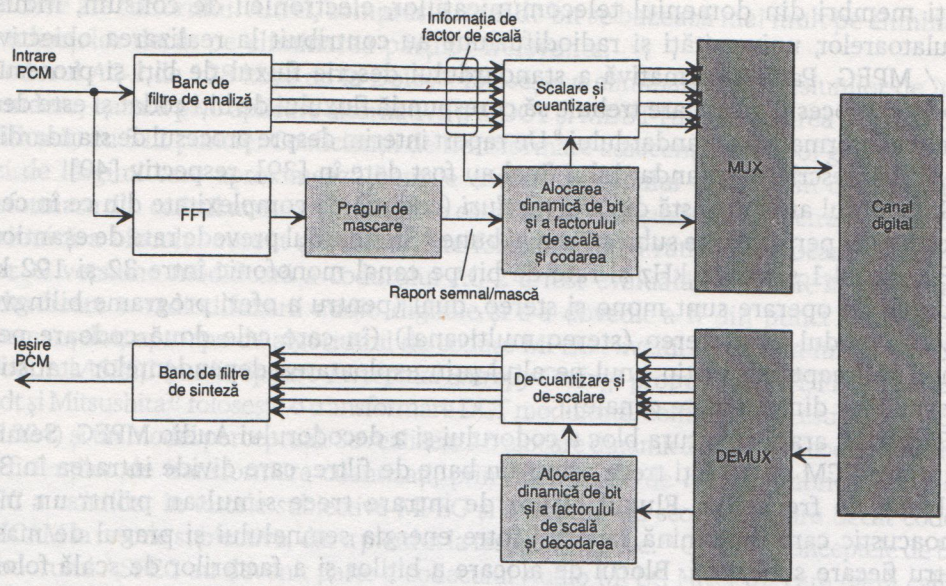


Figura 10. Structura bloc a codorului și decodorului Audio MPEG

iar rata de eșantionare a fiecărei subbenzi decimate este de 48/32 = 1,5 kHz[NT9]. Prin urmare, avem tot atâtea eșantioane ale domeniilor de frecvențe ca și în cazul eșantionării în domeniul timp. Filtrele sunt relativ simple și asigură o bună rezoluție în timp, cu o rezoluție rezonabilă în frecvență.

Proiectarea trebuie să asigure un bun compromis între trei concesii notabile:

- În primul rând, lățimea egală a subbenzilor nu reflectă întocmai comportarea sistemului acustic uman față de frecvențe. Lărgimea unei *benzi critice* ca o funcție de frecvență este un bun indicator al acestei comportări. Figura 1 a comparat deja lărgimile de bandă ale filtrului polifază cu lărgimea benzilor critice. La frecvențe joase, o singură subbandă acoperă mai multe benzi critice. În aceste circumstanțe, numărul biților de cuantizare nu poate fi legat specific de pragul de zgomot disponibil pentru benzile critice individuale. În schimb, banda critică cu mascarea zgomotului cea mai proastă dictează numărul biților de cuantizare necesari pentru întreaga subbandă.
- În al doilea rând, bancul de filtre și inversul său nu reprezintă transformări fără pierderi. Chiar fără cuantizare, transformarea inversă nu poate reface perfect semnalul original. Oricum, prin proiectare, eroarea introdusă de bancul de filtre este extrem de mică și inaudibilă.
- În fine, filtrele cu benzi adiacente au o întrepătrundere majoră în frecvențe. Un semnal de o anumită frecvență poate afecta ieșirile a două filtre adiacente.

Pentru a înțelege bancul de filtre polifază este util să îi analizăm originea. Standardul Audio ISO / MPEG descrie o procedură de calcul a ieșirilor filtrelor polifază, care este similară metodei descrise de Rothweiler.[26] Figura 11 arată structura unui banc de filtre

[NT9] După cum se vede, 1,5 kHz este exact dublul lărgimii de bandă a sub-benzilor care au 750 Hz.

pentru un codor MPEG, bazat pe propunerea lui Rothweiler. Pentru comparație, figura 12 arată diagrama din standardul audio ISO / MPEG pentru același banc de filtre.

Prin combinarea ecuațiilor și a pașilor arătați de organigramă, se poate deduce următoarea ecuație pentru ieșirile bancului de filtre:

$$s_t[i] = \sum_{k=0}^{63} \sum_{j=0}^{7} M[i][k] \times (C[k+64j] \times x[k+64j]) \qquad (6.1)$$

unde:

i este indexul subbenzii și are valori de la 0 la 31

$s_t[i]$ este eșantionul de ieșire al subbenzii i la momentul t, unde t este un multiplu întreg de 32 al intervalelor de eșantionare audio

$C[n]$ este unul din cei 512 coeficienți ai ferestrei de analiză definiți în standard

$x[n]$ un eșantion de intrare citit dintr-un buffer (separator) de 512 eșantioane

$M[i][k] = \cos((2 \times i + 1) \times (k - 16) \times \pi/64)$ sunt coeficienții matricei de analiză.

Ecuația (6.1) este parțial optimizată pentru a reduce numărul de operații. Deoarece funcția dintre paranteze este independentă de valoarea lui i și $M[i][k]$ este independentă de j, ieșirile celor 32 de filtre necesită numai 512+32×64=2560 de înmulțiri și 64×7+32×63 =2464 de adunări, sau circa 80 de înmulțiri și adunări pe ieșire. Reducerea semnificativă în continuare a înmulțirilor și a adunărilor este posibilă. Implementarea eficientă va fi analizată la sfârșitul acestui capitol.

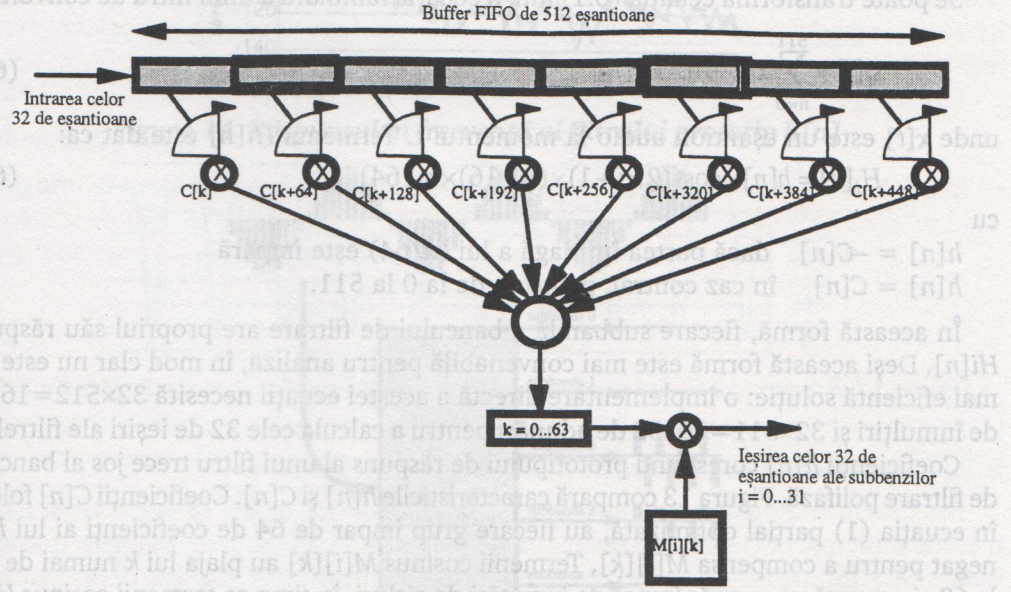


Figura 11. Implementarea bancului de filtre ale codorului MPEG

Remarcați că această implementare a bancului de filtre este eșantionată critic: pentru fiecare 32 de eșantioane de intrare, bancul de filtrare produce 32 de eșantioane de ieșire. De fapt, fiecare dintre cele 32 de filtre de subbandă își sub-eșantionează ieșirea cu 32 pentru a produce o singură ieșire la fiecare 32 de noi eșantioane de intrare.

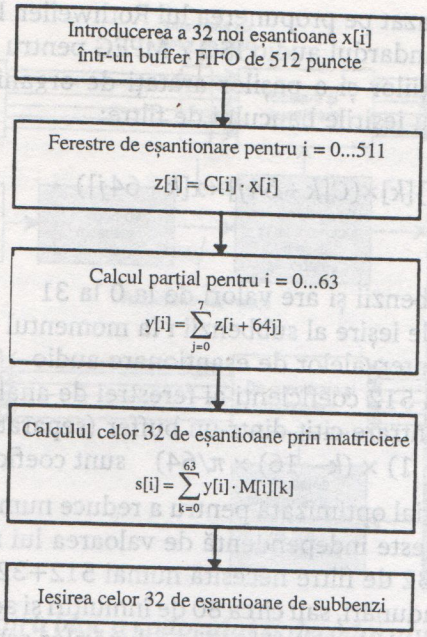


Figura 12. Organigrama

Se poate transforma ecuația (6.1) într-o ecuație familiară a unui filtru de convoluție:

$$s_t[i] = \sum_{n=0}^{511} x[t-n] \times H_i[n] \qquad (6.2)$$

unde $x[t]$ este un eșantion audio la momentul t. Termenul $Hi[n]$ este dat ca:

$$H_i[n] = h[n] \times \cos((2 \times i + 1) \times (n - 16) \times \pi / 64) \qquad (6.3)$$

cu

$h[n] = -C[n]$ dacă partea întreagă a lui $(n/64)$ este impară
$h[n] = C[n]$ în caz contrar, pentru n de la 0 la 511.

În această formă, fiecare subbandă a bancului de filtrare are propriul său răspuns, $Hi[n]$. Deși această formă este mai convenabilă pentru analiză, în mod clar nu este cea mai eficientă soluție: o implementare directă a acestei ecuații necesită $32 \times 512 = 16384$ de înmulțiri și $32 \times 511 = 16352$ de adunări pentru a calcula cele 32 de ieșiri ale filtrelor.

Coeficienții $h[n]$ corespund prototipului de răspuns al unui filtru trece jos al bancului de filtrare polifază. Figura 13 compară caracteristicile $h[n]$ și $C[n]$. Coeficienții $C[n]$ folosiți în ecuația (1) parțial optimizată, au fiecare grup impar de 64 de coeficienți ai lui $h[n]$ negat pentru a compensa $M[i][k]$. Termenii cosinus $M[i][k]$ au plaja lui k numai de la 0 la 63 și acoperă un număr impar de jumătăți de cicluri, în timp ce termenii cosinus $H_i[n]$ au plaja lui n de la 0 la 511 și acoperă de opt ori numărul de jumătăți de ciclu.

Ecuația (6.3) pentru $H_i[n]$ arată clar că fiecare este o modulație cu un termen cosinus a prototipului răspunsului, pentru a deplasa răspunsul trece jos la banda de frecvență potrivită, de aici și denumirea de *filtre polifază*. Aceste filtre au frecvențele centrale la multipli impari ai lui $\pi/(64T)$ unde T este perioada de eșantionare audio și fiecare are o lățime de bandă nominală de $\pi/(32T)$.

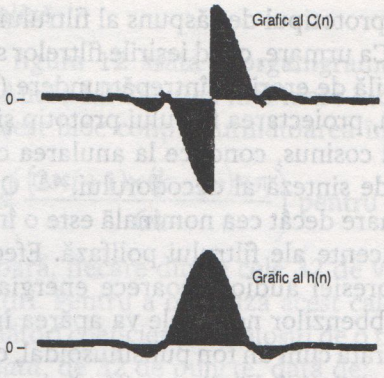


Figura 13. Comparație între h[n] și C[n]

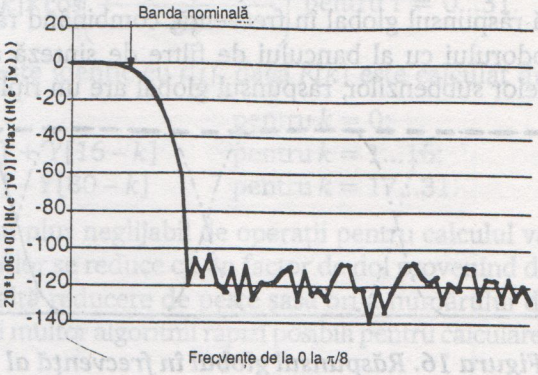


Figura 14. Răspunsul în frecvență al filtrului prototip h[n]

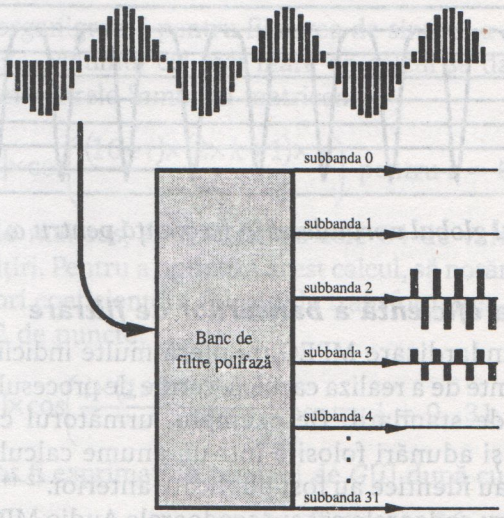


Figura 15. Întrepătrunderea (aliasing): intrări sinusoidale pure pot produce ieșiri nenule pentru două subbenzi. Intrare: undă sinus 1,5 kHz, eșantionată la 32 kHz; arătate 64 din 256 de eșantioane. Ieșire: 8×32 de eșantioane de subbandă; ambele subbenzi 2 și 3 au valori de ieșire semnificative.

Aşa cum arată figura 14, prototipul de răspuns al filtrului nu are o margine abruptă la capetele benzii nominale. Ca urmare, când ieşirile filtrelor sunt sub-eşantionate cu 32, există o cantitate considerabilă de erori de întrepătrundere (*aliasing*), cum s-a discutat deja în capitolul 4.2. Oricum, proiectarea filtrului prototip şi includerea unei deplasări de fază potrivite în termenii cosinus, conduce la anularea completă a acestor erori la ieşirile bancului de filtrare de sinteză al decodorului.[26,41] O altă consecinţă a folosirii unui filtru cu o bandă mai mare decât cea nominală este o întrepătrundere a acoperirii în frecvenţă a benzilor adiacente ale filtrului polifază. Efectul se poate manifesta în detrimentul eficienţei compresiei audio, deoarece energia semnalului în zonele de vecinătate ale marginilor subbenzilor nominale va apărea în două ieşiri adiacente ale filtrului polifază. Figura 15 arată cum un ton pur sinusoidal, care are energie numai pe o singură frecvenţă, apare la ieşirile a două filtre polifază.

Deşi bancul de filtrare polifază nu este fără pierderi, orice eroare implicată este mică. Figurile 16 şi 17 arată răspunsul global în frecvenţă, combinând răspunsul bancului de filtre de analiză al codorului cu al bancului de filtre de sinteză al decodorului. Fără cuantizarea eşantioanelor subbenzilor, răspunsul global are un riplu de numai 0,7 dB.

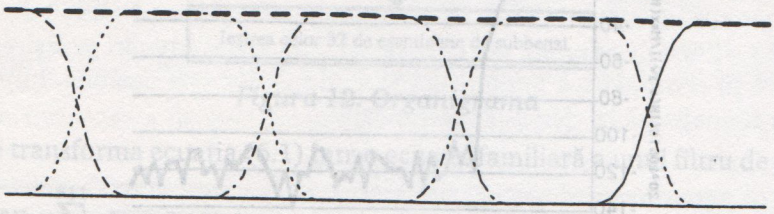


Figura 16. Răspunsul global în frecvenţă al bancului de filtre polifază pentru trei subbenzi

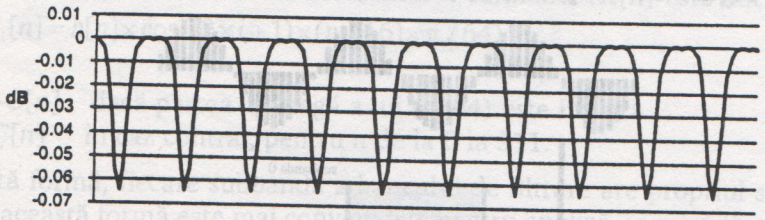


Figura 17. Răspunsul global normalizat în frecvenţă pentru $\omega = 0...5\pi/16$ (în dB)

6.2. *Implementarea eficientă a bancurilor de filtrare*

Documentul de standardizare MPEG nu oferă multe indicii în acest sens. Există metode mult mai eficiente de a realiza caculele cerute de procesul de codare şi decodare decât cele înfăţişate de standard. De exemplu, următorul capitol detaliază cum numărul de înmulţiri şi adunări folosite într-un anume calcul poate fi redus de 12 ori. Metode similare sau identice au fost publicate anterior.[42-44] Multe alte optimizări sunt posibile atât pentru codoarele cât şi decodoarele Audio MPEG. Întrucât mişcarea datelor poate să ia în anumite arhitecturi de procesoare tot atâta timp cât înmulţirile şi adunările, tehnicile descrise mai jos ar putea să nu fie cele mai eficiente pentru toate configuraţiile hardware.

Bancul de filtrare de analiză

În capitolul precedent, figura 12 arăta o organigramă a filtrului de subbenzi de analiză folosit de codorul Audio MPEG. Cea mai mare încărcătură de calcul este dată de antepenultimul bloc. Acest bloc conține următoarea înmulțire matriceală:

$$S[i]=\sum_{k=0}^{63} Y[k]\times\cos\left(\frac{(2\times i+1)\times(k-16)\times\pi}{64}\right) \text{ pentru } i = 0...31. \qquad (6.4)$$

Folosind ecuația anterioară, fiecare dintre cele 32 de valori ale lui $S[i]$ necesită 63 de adunări și 64 de înmulțiri. Pentru a optimiza acest calcul, să notăm că coeficienții $M[i][k]$ sunt foarte asemănători coeficienților folosiți de o transformare cosinus discretă (DCT) inversă *ne-normalizată*, de 32 de puncte, dată de:

$$f[i]=\sum_{k=0}^{31} F[k]\times\cos\left(\frac{(2\times i+1)\times k\times\pi}{64}\right) \text{ pentru } i = 0...31. \qquad (6.5)$$

Într-adevăr, $S[i]$ este identic cu $f[i]$, dacă $F[k]$ este calculat după cum urmează:

$$F[k] = \begin{cases} Y[16] & \text{pentru } k = 0; \\ Y[k+16] + Y[16-k] & \text{pentru } k = 1...16; \\ Y[k+16] - Y[80-k] & \text{pentru } k = 17...31. \end{cases} \qquad (6.6)$$

Prin urmare, cu un plus neglijabil de operații pentru calculul valorilor $F[k]$, numărul înmulțirilor și adunărilor se reduce cu un factor de doi provenind din înjumătățirea plajei în care variază k. O altă reducere de peste șase ori a numărului de adunări și înmulțiri apare datorită folosirii multor algoritmi rapizi posibili pentru calcularea transformării inverse DCT.[45-47]

Bancul de filtrare de sinteză

Figura 18 arată o organigramă pentru filtrarea de sinteză a subbenzilor, folosită de decodorul Audio MPEG. Volumul cel mai mare de calcul se datorează primului bloc. Acest bloc conține următoarele înmulțiri matriceale:

$$V[i]=\sum_{k=0}^{31} S[k]\times\cos\left(\frac{(16+i)\times(2\times k+1)\times\pi}{64}\right) \text{ pentru } i = 0...63. \qquad (6.7)$$

Folosind ecuația de mai sus, fiecare dintre cele 64 de valori $V[i]$ necesită 31 de adunări și 32 de înmulțiri. Pentru a optimiza acest calcul, să notăm că coeficienții $N[i][k]$ sunt foarte asemănători coeficienților folosiți de transformarea cosinus discretă (DCT) ne-normalizată, de 32 de puncte:

$$G[i]=\sum_{k=0}^{31} S[k]\times\cos\left(\frac{i\times(2\times k+1)\times\pi}{64}\right) \text{ pentru } i = 0...31. \qquad (6.8)$$

Într-adevăr, $V[i]$ pot fi exprimați în termeni de $G[i]$ după cum urmează:

$$V[i] = \begin{cases} G[i+16] & \text{pentru } 0 <= i < 16; \\ 0 & \text{pentru } i = 16; \\ -G[48-i] & \text{pentru } 16 < i < 49; \\ -G[i-48] & \text{pentru } 49 <= i < 64. \end{cases} \qquad (6.9)$$

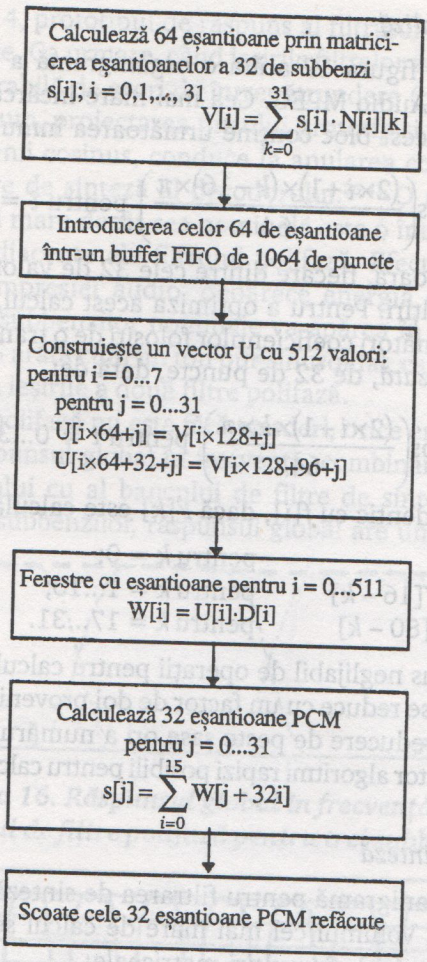


Figura 18. Organigrama bancului de filtrare al decodorului Audio MPEG

Ca și în cazul bancului de filtrare de analiză, această expresie înjumătățește numărul de calcule necesare și o altă reducere de peste șase ori a înmulțirilor și adunărilor se obține prin folosirea unuia din mulții algoritmi rapizi posibili pentru calculul DCT.

6.3. *Modelul psihoacustic*

Modelul psihoacustic analizează semnalul audio și calculează posibilitățile de mascare a zgomotului funcție de frecvență.[48-51] Posibilitățile de mascare a unei componente a unui semnal dat depind de poziția frecvenței, de tăria ei și de tonalitate. Codorul folosește această informație pentru a decide cum să reprezinte mai bine semnalul audio de intrare, folosind numărul lui limitat de biți disponibili. Standardul de codare Audio MPEG oferă două exemple de implementare a modelelor psihoacustice. Modelul psihoacustic 1 este mai puțin complex decât modelul psihoacustic 2 și acceptă într-un fel, mai multe compromisuri pentru a face calculul mai simplu. Ambele modele se pretează pentru oricare dintre nivelurile de compresie. Totuși, numai modelul 2 include modificări specifice pentru a se preta Nivelului III.

Există o libertate considerabilă în a implementa modelul psihoacustic. Necesitatea respectării întocmai a modelului depinde de factorul de compresie pe care îl dorim și de aplicația pe care intenționăm să o realizăm. Pentru niveluri mici de compresie, unde există o dispónibilitate generoasă de biți de codare, se poate folosi o scurtcircuitare completă a modelului psihoacustic. În acest caz, procesul de alocare de biți poate distribui iterativ biți subbenzilor cu raportul semnal / zgomot cel mai redus, în concordanță cu tehnica folosită pentru convenționalul ATC.[20,22]

Sunt subliniați mai jos *pașii de bază implicați în calculele psihoacustice* pentru fiecare model. Diferențele între cele două modele vor fi evidențiate.

- *Alinierea în timp a datelor audio (sincronizarea)*. Există o evaluare psihoacustică pe cadru. Datele audio trimise modelului psihoacustic trebuie să fie corelate în timp cu datele audio care trebuie codate. Modelul psihoacustic trebuie să țină cont atât de timpii de întârziere prin bancul de filtrare cât și de un offset al datelor, necesar pentru ca datele relevante să fie centrate în fereastra psihoacustică analizată. De exemplu, când se folosește modelul psihoacustic 1 pentru Nivelul I, întârzierea prin bancul de filtrare este de 256 de eșantioane și offsetul necesar centrării a 384 de eșantioane ale unui cadru al Nivelului I în cele 512 puncte ale ferestrei analizate este (512 – 384)/2 = 64 de puncte. Offsetul net este 320 de puncte pentru a alinia în timp datele modelului psihoacustic cu ieșirile bancului de filtrare.
- *Conversia audio la o reprezentare în domeniul frecvență*. Modelul psihoacustic ar trebui să folosească o transformare timp-frecvență separată, independentă, în locul bancului de filtrare polifază, deoarece are nevoie de o rezoluție mai fină în frecvență pentru un calcul precis al pragului de mascare. Ambele modele psihoacustice folosesc o transformare Fouriér în acest plan. O ponderare standard Hann, aplicată datelor audio înainte de transformarea Fourier, condiționează reducerea efectelor de margine ale datelor în fereastra transformată.

Modelul psihoacustic 1 folosește o fereastră de analiză de 512 eșantioane pentru Nivelul I și 1024 eșantioane pentru Nivelurile II și III. Pentru că sunt numai 384 de eșantioane într-un cadru al Nivelului I, o fereastră de 512 eșantioane asigură o acoperire adecvată. Aici dimensiunea redusă a ferestrei reduce efortul de calcul. Nivelurile II și III folosesc cadre de 1152 eșantioane, astfel că o fereastră de 1024 de eșantioane nu asigură o acoperire totală. În timp ce o fereastră de analiză ideală ar trebui să acopere în întregime eșantioanele care trebuie codate, o fereastră de 1024 eșantioane este un compromis rezonabil. Eșantioanele care cad în afara ferestrei de analiză, în general nu vor avea un impact major asupra evaluării psihoacustice.

Modelul psihoacustic 2 folosește o fereastră de 1024 eșantioane pentru toate nivelurile. Pentru Nivelul I, modelul centrează cele 384 de eșlantioane ale cadrului în fereastra psihoacustică așa cum am discutat mai înainte. Pentru Nivelurile II și III, modelul face două calcule de 1024 de puncte psihoacustice pentru fiecare cadru. Primul calcul centrează prima jumătate a celor 1152 eșantioane în fereastra de analiză, iar al doilea calcul centrează cealaltă jumătate. Modelul combină cele două rezultate ale calculelor folosind raportul semnal / mască cel mai mare dintre cele două pentru fiecare subbandă. Astfel se selectează pragul de mascare cel mai mic dintre cele două, pentru fiecare subbandă.

- *Procesarea valorilor spectrale în grupuri legate de lărgimile benzilor critice.* Pentru a simplifica într-o măsură calculele psihoacustice, ambele modele procesează valorile frecvențelor în cuante perceptuale.
- *Separarea valorilor spectrale în componente tonale și non-tonale.* Ambele modele identifică și separă componentele tonale și cele de tip zgomot din semnalul audio deoarece abilitățile de mascare pentru cele două componente sunt diferite.

Modelul psihoacustic 1 identifică componentele tonale pe baza vârfurilor locale ale spectrului de putere audio. După procesarea componentelor tonale, modelul 1 sumează valorile spectrale rămase, într-o singură componentă non-tonală pe fiecare bandă critică. Indexul frecvenței fiecăreia dintre aceste componente concentrate non-tonale este valoarea cea mai apropiată de media geometrică a benzii critice înconjurătoare.

Modelul psihoacustic 2 nu separă niciodată componentele tonale de cele non-tonale. În schimb, el calculează un index al tonalității ca o funcție de frecvență. Acest index dă o măsură a asemănării componentei cu un ton sau cu un zgomot. Modelul 2 folosește acest index pentru o interpolare a valorilor de ton-care-maschează-zgomot și de zgomot-care-maschează-ton. Indexul tonalității se bazează pe o măsură a predictibilității. Modelul 2 folosește datele din cele două ferestre de analiză precedente pentru a prezice, printr-o extrapolare liniară, valorile componentelor pentru fereastra curentă. Componentele tonale sunt mai predictibile și, de aceea, vor avea indici de tonalitate mai ridicați. Deoarece acest proces se bazează pe mai multe date, este de așteptat să departajeze mai bine decât modelul 1 componentele tonale și non-tonale.

- *Aplicarea unei funcții de împrăștiere.* Abilitățile de mascare ale unui semnal dat se extind și asupra benzilor critice înconjurătoare. Modelul determină pragurile de mascare a zgomotului prin aplicarea întâi a unei mascări determinate empiric (modelul 1) sau a unei funcții de împrăștiere (modelul 2) asupra componentelor semnalului.
- *Determinarea unei limite minime a valorilor pragurilor.* Ambele modele includ un prag de mascare absolut, determinat empiric, pragul de liniște. Acest prag este limita minimă de audibilitate a sunetului.
- *Găsirea pragului de mascare pentru fiecare subbandă.* Ambele modele psihoacustice calculează pragurile de mascare cu o rezoluție în frecvență mai bună decât cea oferită de bancul de filtre polifază. Ambele modele trebuie să determine o valoare de prag a subbenzii dintr-o multitudine de praguri de mascare calculate pentru frecvențe din acea subbandă.

Modelul psihoacustic 1 selectează pragul minim de mascare din fiecare subbandă. În timp ce această abordare este bună pentru benzile de frecvențe joase unde subbenzile sunt înguste comparativ cu benzile critice, poate fi imprecisă pentru subbenzile de frecvențe înalte deoarece benzile critice pentru aceste game de frecvențe acoperă câteva subbenzi. Aceste imprecizii apar deoarece modelul 1 concentrează toate componentele non-tonale din fiecare bandă critică într-o singură valoare la o singură frecvență. De fapt, modelul 1 convertește componentele non-tonale într-o formă de componentă tonală. O subbandă dintr-o bandă critică mai largă, dar departe de componenta concentrată non-tonală, nu va primi o decizie de mascare non-tonală potrivită. Această abordare este un compromis pentru a reduce volumul calculelor.

Modelul psihoacustic 2 selectează minimul pragurilor de mascare acoperite de subbandă numai când banda este largă față de banda critică din acea zonă de frecvențe. Acolo unde banda este îngustă față de banda critică, se folosește media pragurilor de mascare acoperite de subbandă. Modelul 2 nu este mai puțin precis în zonele de frecvență înaltă deoarece nu se concentrează pe componentele non-tonale.

- *Calcularea raportului semnal / mască.* Modelul psihoacustic calculează raportul semnal / mască ca raport între energia semnalului din subbandă (sau pentru Nivelul III din grupul de subbenzi) și minimul pragului de mascare pentru acea subbandă. Modelul va trece această informație (valoare) către etajul de alocare a biților (sau a zgomotului) din cadrul codorului.

6.4. *Un exemplu de analiză realizată de modelul psihoacustic*

Acest capitol dă un exemplu ilustrativ al analizei folosite de modelele psihoacustice 1 și 2. Figura 19 este reprezentarea spectrală a unui exemplu de semnal audio care trebuie analizat psihoacustic și comprimat. Semnalul constă dintr-o combinație între un ton sinusoidal puternic de 11,250 kHz și zgomot de frecvențe joase.

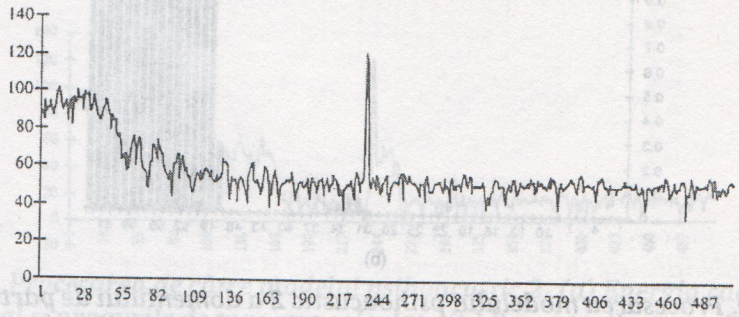


Figura 19. Energia la intrarea unui cadru audio în funcție de frecvență. Zgomot de frecvență joasă și o undă de 11250 Hz sinus, eșantionate la 48 kHz

6.4.1. Exemplu pentru modelul psihoacustic 2

Procesările folosite de modelul psihoacustic 2 sunt oarecum mai ușor de vizualizat, așa că acest model va fi descris mai întâi. Figura 20(a) arată rezultatul, în concordanță cu modelul psihoacustic 2, al transformatei semnalului audio în domeniul perceptual (partiție în 63 de treimi de benzi critice) și aplicării apoi a funcției de împrăștiere. De notat deplasarea vârfului sinusoidal și expandarea zgomotului de bandă joasă în comparație cu figura 19. Transformarea perceptuală extinde regiunea frecvențelor joase și comprimă regiunea frecvențelor înalte. Deoarece funcția de împrăștiere este aplicată într-un domeniu perceptual, forma funcției de înprăștiere este relativ uniformă, ca o funcție de partiție. Figura 21 arată un grafic al funcției de împrăștiere. Figura 20(b) arată indexul tonalităților pentru cadrul audio, așa cum a fost calculat după modelul psihoacustic 2. Figura 22(a) arată un grafic al pragurilor de mascare, așa cum au fost calculate de model pe baza energiei de împrăștiere și a indexului de tonalități. Figura cuprinde graficele pragurilor de mascare atât înainte cât și după încorporarea pragului de liniște, pentru a vedea impactul acestuia. De notat că pragul de liniște mărește semnificativ pragul de mascare al zgomotului pentru frecvențele înalte. Sistemul auditiv

uman este mult mai puțin sensibil în această regiune (vezi figura 2). Totodată să observăm cum mărește semnalul sinusoidal pragul de mascare pentru frecvențele învecinate. Pragul de mascare este calculat în domeniul uniform al frecvențelor în locul domeniului perceptual, în pregătirea pasului final al modelului psihoacustic, calcularea raportului semnal / mască (SMR) pentru fiecare subbandă.

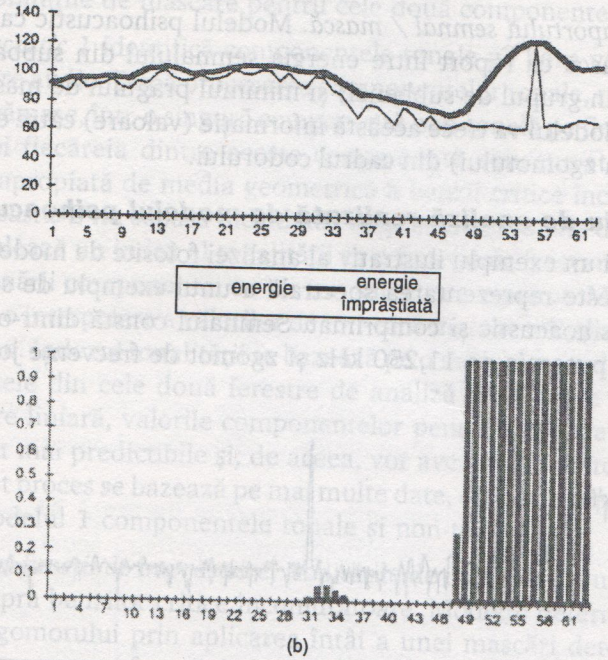


Figura 20. Procesarea modelului psihoacustic 2 a domeniului de partiție (cu o partiție de 63 de treimi de benzi critice). (a) Energia audio și energia de împrăștiere în domeniul perceptual (b) Indexul tonalităților

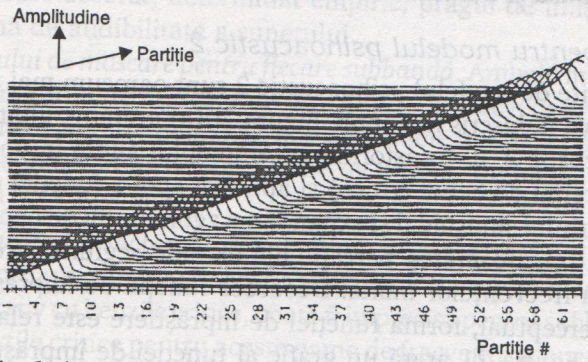


Figura 21. Funcția de împrăștiere pentru fiecare partiție (modelul psihoacustic 2) (cu o partiție de 63 de treimi de subbenzi critice).

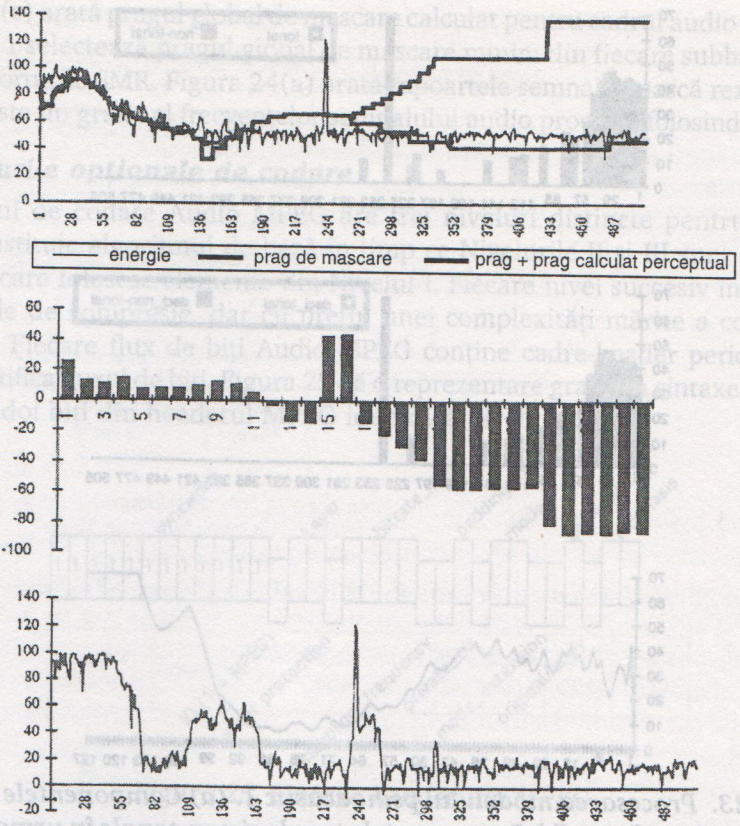


Figura 22. Procesarea de către modelul psihoacustic 2. (a) Energia cadrului audio original și pragurile de mascare calculate (b) Raporturile semnal / mască (c) Energia cadrului audio codat (64 kbps)

Figura 22(b) este un grafic al acestor rezultate iar figura 22(c) este un grafic al cadrului audio procesat folosind aceste rapoarte SMR. În acest exemplu compresia audio a fost severă (de la 768 la 64 kbps) astfel încât codorul s-ar putea să nu fie capabil să mascheze în întregime zgomotul de cuantizare.

Exemplu pentru modelul psihoacustic 1

Acest exemplu folosește același cadru audio ca și cel precedent. Figura 23(a) arată cum identifică modelul psihoacustic 1 vârfurile spectrale locale drept componente tonale sau non-tonale. Figura 23(b) arată componentele tonale și non-tonale rămase în urma procesului de decimare. Acest proces elimină atât componentele care se află sub pragul de liniște precum și componentele tonale mai slabe din interiorul unei aproximativ jumătăți de bandă critică (0,5 Bark), în care există o componentă tonală puternică. Modelul psihoacustic 1 folosește componentele tonale și non-tonale decimate pentru a determina pragul de mascare global într-un domeniu de frecvențe sub-eșantionate. Domeniul frecvențelor sub-eșantionate corespunde aproximativ unui domeniu perceptual.

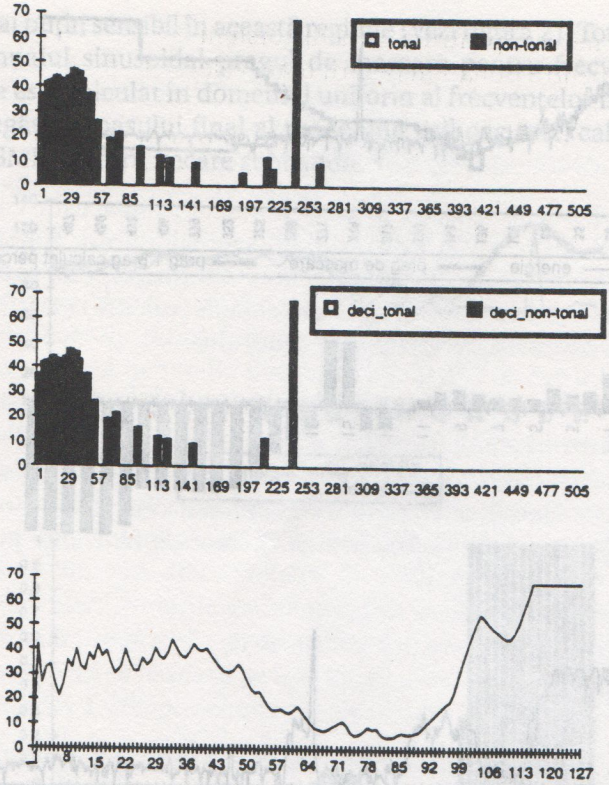


Figura 23. Procesarea modelului psihoacustic 1. (a) Componentele tonale și non-tonale identificate (b) Componentele tonale și non-tonale în urma decimării (c) Pragurile globale de mascare în spațiul frecvențelor sub-eșantionate

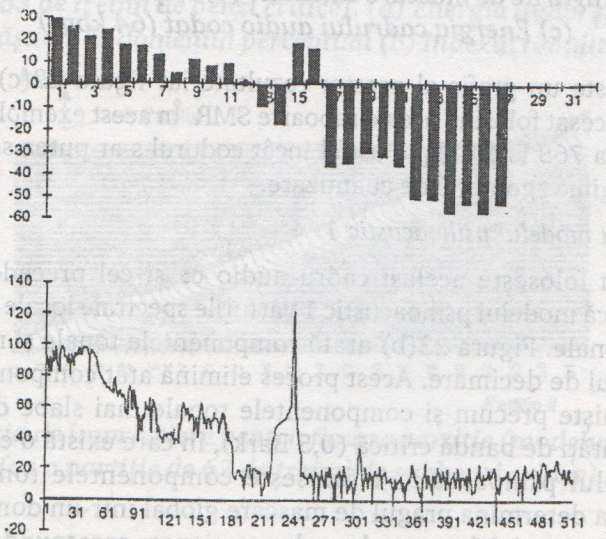


Figura 24. Rezultatul procesării cu modelul psihoacustic 1. (a) Raportul semnal / mască (b) Energia cadrului audio codat (64 kbps)

Figura 23(c) arată pragul global de mascare calculat pentru cadrul audio dat. Modelul psihoacustic 1 selectează pragul global de mascare minim din fiecare subbandă, pentru a calcula raporturile SMR. Figura 24(a) arată rapoartele semnal / mască rezultate, iar figura 24(b) este un grafic al frecvențelor semnalului audio procesat folosind aceste SMR.

6.5. *Nivelurile opționale de codare*

Standardul de codare Audio MPEG are trei niveluri distincte pentru compresie. Nivelul I constituie algoritmul de bază în timp ce Nivelurile II și III sunt îmbunătățiri (dezvoltări) care folosesc elemente din Nivelul I. Fiecare nivel succesiv îmbunătățește performanțele de compresie, dar cu prețul unei complexități mărite a codorului și a decodorului. Fiecare flux de biți Audio MPEG conține cadre-header periodic spațiate pentru a identifica fluxul de biți. Figura 25 dă o reprezentare grafică a sintaxei headerului. Un câmp de doi biți din headerul MPEG identifică nivelul folosit.

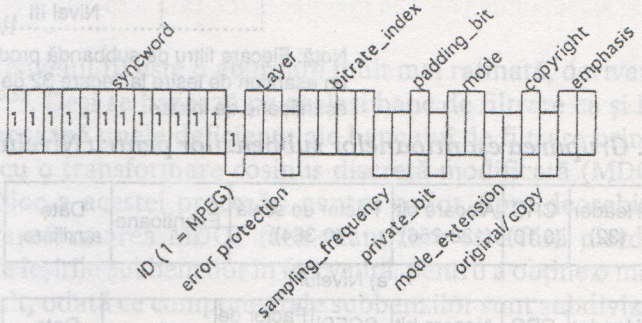


Figura 25. Sintaxa headerului MPEG

6.5.1. *Nivelul I*

Algoritmul Nivelului I codează semnalul audio în cadre de 384 de eșantioane. El face aceasta prin gruparea a câte 12 eșantioane din fiecare din cele 32 de subbenzi, așa cum se arată în figura 26. În afara codului pentru datele audio, fiecare cadru conține un header, opțional un cuvânt de verificare a erorilor CRC (*Cyclic Redundancy Code*) și posibile date auxiliare. Figura 27(a) arată aranjarea acestor date în fluxul de biți al Nivelului I. Numerele din paranteze arată numărul posibil de biți care pot fi utilizați pentru codarea fiecărui câmp. Fiecare grup de 12 eșantioane primește o alocare de biți și dacă alocarea de biți nu este zero, primește și un factor de scală. Alocarea de biți spune decodorului numărul de biți care s-au folosit pentru reprezentarea fiecărui eșantion. Pentru Nivelul I acest număr poate fi de la 0 la 15 pe subbandă. Factorul de scală este un multiplicator care dimensionează eșantioanele astfel încât să facă uz total de plaja de cuantizare. Fiecare factor de scală are 6 biți. Decodorul înmulțește ieșirea cuantizată decodată cu factorul de scală, pentru a reface valoarea cuantizată a subbenzii. Plaja dinamică a factorului de scală este mai mare de 120 dB. Combinația dintre numărul de biți alocați și factorul de scală asigură un potențial de reprezentare a eșantioanelor cu o dinamică mult peste 120 dB. Codarea stereo multicanal alterează ușor reprezentarea eșantioanelor canalelor audio stânga și dreapta și va fi descrisă mai târziu.

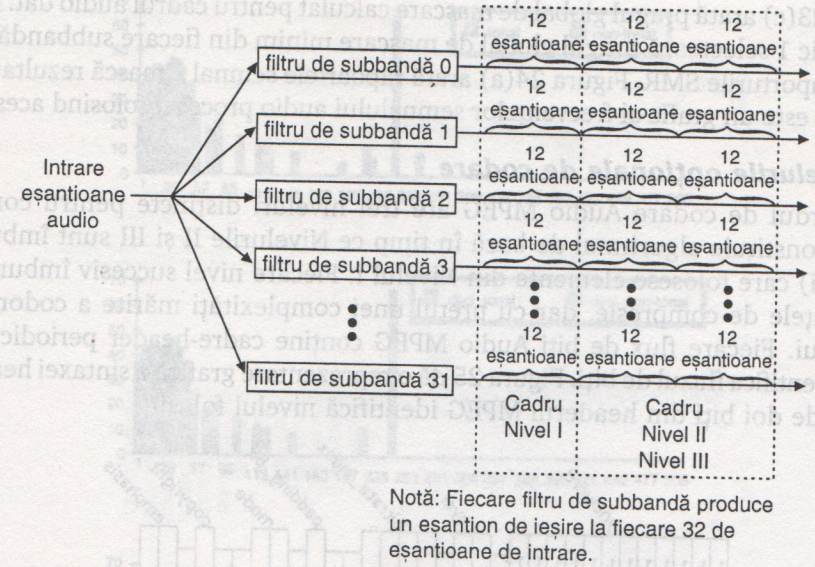


Figura 26. Gruparea eșantioanelor subbenzilor pentru Nivelurile I și II

Header (32)	CRC (0.16)	Alocare bit (128-256)	Factor de scală (0-384)	Eșantioane	Date auxiliare

a) Nivelul I

Header (32)	CRC (0.16)	Alocare bit (128-256)	SCFSI (0-60)	Factor de scală (0-1080)	Eșantioane	Date auxiliare

b) Nivelul II

Header (32)	CRC (0.16)	Informații laterale (136.256)	Datele principale; nu neapărat legate de acest cadru. Vezi fig.30

c) Nivelul III

Figura 27. Formatele cadrelor celor trei niveluri MPEG (a) Nivelul I (b) Nivelul II (c) Nivelul III

6.5.2. *Nivelul II*

Algoritmul Nivelului II este o dezvoltare a Nivelului I. El codează datele audio în grupe mai mari și impune unele restricții asupra posibilităților de alocare de bit pentru valorile din benzile de la frecvențe medii în sus. De asemenea, el reprezintă alocarea de bit, factorul de scală și eșantioanele cuantizate, cu un cod mai compact. Nivelul II oferă o calitate audio mai bună prin salvarea biților din această zonă, astfel încât mai mulți biți rămân disponibili pentru reprezentarea valorilor subbenzilor.

Codorul de Nivelul II formează cadre de 1152 de eșantioane pe canal audio. În timp ce Nivelul I codează datele într-un singur grup de 12 eșantiaone pentru fiecare subbandă, Nivelul II codează datele în trei grupe de 12 eșantioane pentru fiecare subbandă. Figura 26 arată și acest mod de grupare. Din nou, lăsând la o parte redundanța codării stereo, există o alocare de bit și până la trei factori de scală pentru fiecare grup de 12 eșantioane din trio. Codorul folosește factori de scală diferiți pentru cele trei

grupe de 12 eşantioane, numai dacă acest lucru este necesar pentru a evita distorsiuni audibile. Codorul poate folosi acelaşi factor de scală pentru două sau pentru toate cele trei grupe în alte două cazuri. Unul, când valorile factorilor de scală sunt suficient de apropiate. Doi, când codorul anticipează că mascarea temporală a zgomotului de către sistemul auditiv uman va ascunde distorsiunile cauzate de folosirea unui singur factor de scală în loc de doi sau trei. Câmpul informaţiei de selectare a factorului de scală (SCFSI = *Scale Factor Selection Information*) din fluxul de biţi al Nivelului II informează decodorul dacă şi cum să împartă valorile factorilor de scală. Figura 27(b) arată aranjarea diferitelor câmpuri de date într-un flux de Nivelul II.

O altă îmbunătăţire se realizează pentru situaţiile în care codorul de Nivelul II alocă 3, 5 sau 9 niveluri pentru cuantizarea subbenzilor. În aceste circumstanţe, codorul de Nivelul II reprezintă trei valori consecutive cuantizate cu un singur cuvânt de cod, mai compact.

6.5.3. Nivelul III

Algoritmul de Nivelul III este o abordare mult mai rafinată, derivată din algoritmii ASPEC şi OCF[37,50,52]. Deşi se bazează pe acelaşi banc de filtrare ca şi Nivelurile I şi II, Nivelul III compensează unele deficienţe ale bancului de filtrare printr-o procesare a ieşirilor filtrelor cu o transformare cosinus discretă modificată (MDCT).[27] Figura 28 arată o schemă bloc a acestei procesări pentru codor. Spre deosebire de bancul de filtre polifază, transformarea MDCT fără cuantizare este fără pierderi. MDCT subdivide mai departe ieşirile subbenzilor în frecvenţă, pentru a obţine o mai bună rezoluţie spectrală. Mai mult, odată ce componentele subbenzilor sunt subdivizate în frecvenţă, codorul de Nivelul III poate să anuleze parţial unele distorsiuni de întrepătrundere (*aliasing*) cauzate de bancul de filtrare polifază. Desigur, decodorul de Nivelul III trebuie să refacă întrepătrunderile respective, astfel ca MDCT inversă să poată reconstrui eşantioanele subbenzilor în forma lor originală, distorsionate, pentru bancul de filtrare pentru sinteză.

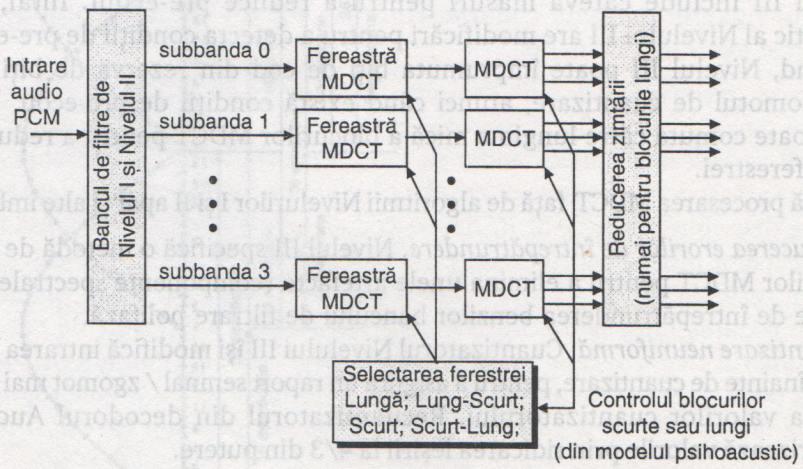


Figura 28. Procesarea bancului de filtre audio Nivelul III MPEG (partea de codor)

Nivelul III specifică două lungimi de bloc MDCT: un bloc lung de 18 eşantioane sau un bloc scurt de 6. Există o întrepătrundere de 50% între ferestrele succesive ale transformatei, astfel că dimensiunile ferestrelor sunt de 36, respectiv 12. Lungimea mare a blocurilor permite o mai bună rezoluţie pentru semnalele audio cu caracteristici staţionare, în timp ce lungimea mică a blocurilor oferă o mai bună rezoluţie a tranziţiilor.[29] De observat că lungimea blocurilor scurte este o treime din lungimea blocurilor lungi. În cazul lucrului cu blocuri mici, trei blocuri mici înlocuiesc un bloc mare, astfel că numărul eşantioanelor MDCT pentru un cadru de eşantioane audio este neschimbat faţă de selecţia lungimii blocurilor. Pentru un cadru dat de eşantioane audio, transformatele MDCT pot avea toate aceeaşi lungime de bloc (lung sau scurt), sau pot cuprinde blocuri mixate. În cazul blocurilor mixate, MDCT pentru cele două subbenzi cu frecvenţele cele mai joase sunt folosite blocuri lungi, iar MDCT pentru celelalte 30 de subbenzi de frecvenţe mai înalte sunt folosite blocuri scurte. Acest mod asigură o mai bună rezoluţie în frecvenţă pentru frecvenţele joase, unde este cea mai necesară, fără a sacrifica rezoluţia în timp la frecvenţe mari.

Comutarea între lungimi mici şi mari ale blocurilor nu este instantanee. Un bloc lung cu date specializate pentru tranziţie scurt-spre-lung sau lung-spre-scurt este intercalat între cele două tipuri. Figura 29 arată cum asigură ferestrele MDCT tranzitarea între modurile de lucru cu blocuri lungi şi scurte.

Deoarece procesarea MDCT a unei subbenzi de semnal asigură o rezoluţie mai bună în frecvenţă, ea are în consecinţă o rezoluţie mai slabă în timp. MDCT operează pe 12 sau 36 eşantioane ale filtrelor polifază, astfel că fereastra efectivă de timp a eşantioanelor audio implicate în această procesare este de 12 sau de 36 de ori mai largă. Cuantizarea valorilor MDCT va cauza erori care se vor întinde peste această fereastră de timp mai largă şi este mai probabil ca aceste erori să producă distorsiuni audibile. Aceste distorsiuni se manifestă de obicei ca pre-ecouri, deoarece mascarea temporală a zgomotului care începe înaintea unui semnal dat, este mai slabă decât mascarea zgomotului de după semnal (vezi capitolele 3.2 şi 4.2).

Nivelul III include câteva măsuri pentru a reduce pre-ecoul. Întâi, modelul psihoacustic al Nivelului III are modificări pentru a detecta condiţii de pre-ecou. În al doilea rând, Nivelul III poate împrumuta biţi de cod din rezerva de biţi pentru a reduce zgomotul de cuantizare, atunci când există condiţii de pre-ecou. În sfârşit, codorul poate comuta către lungime mică a blocurilor MDCT pentru a reduce timpul efectiv al ferestrei.

Pe lângă procesarea MDCT, faţă de algoritmii Nivelurilor I şi II apar şi alte îmbunătăţiri:

- *Reducerea erorilor de întrepătrundere*. Nivelul III specifică o metodă de procesare a valorilor MDCT pentru a elimina unele artefacte (componente spectrale parazite) cauzate de întrepătrunderea benzilor bancului de filtrare polifază.
- *Cuantizare neuniformă*. Cuantizatorul Nivelului III îşi modifică intrarea la 3/4 din putere înainte de cuantizare, pentru a asigura un raport semnal / zgomot mai consistent în plaja valorilor cuantizatorului. Recuantizatorul din decodorul Audio MPEG reliniarizează valorile prin ridicarea ieşirii la 4/3 din putere.
- *Benzi de factori de scală*. Spre deosebire de Nivelurile I şi II, unde pot fi factori de scală diferiţi pentru fiecare subbandă, Nivelul III foloseşte benzi de factori de scală. Aceste benzi ocupă câţiva coeficienţi MDCT şi au aproximativ lăţimea unei

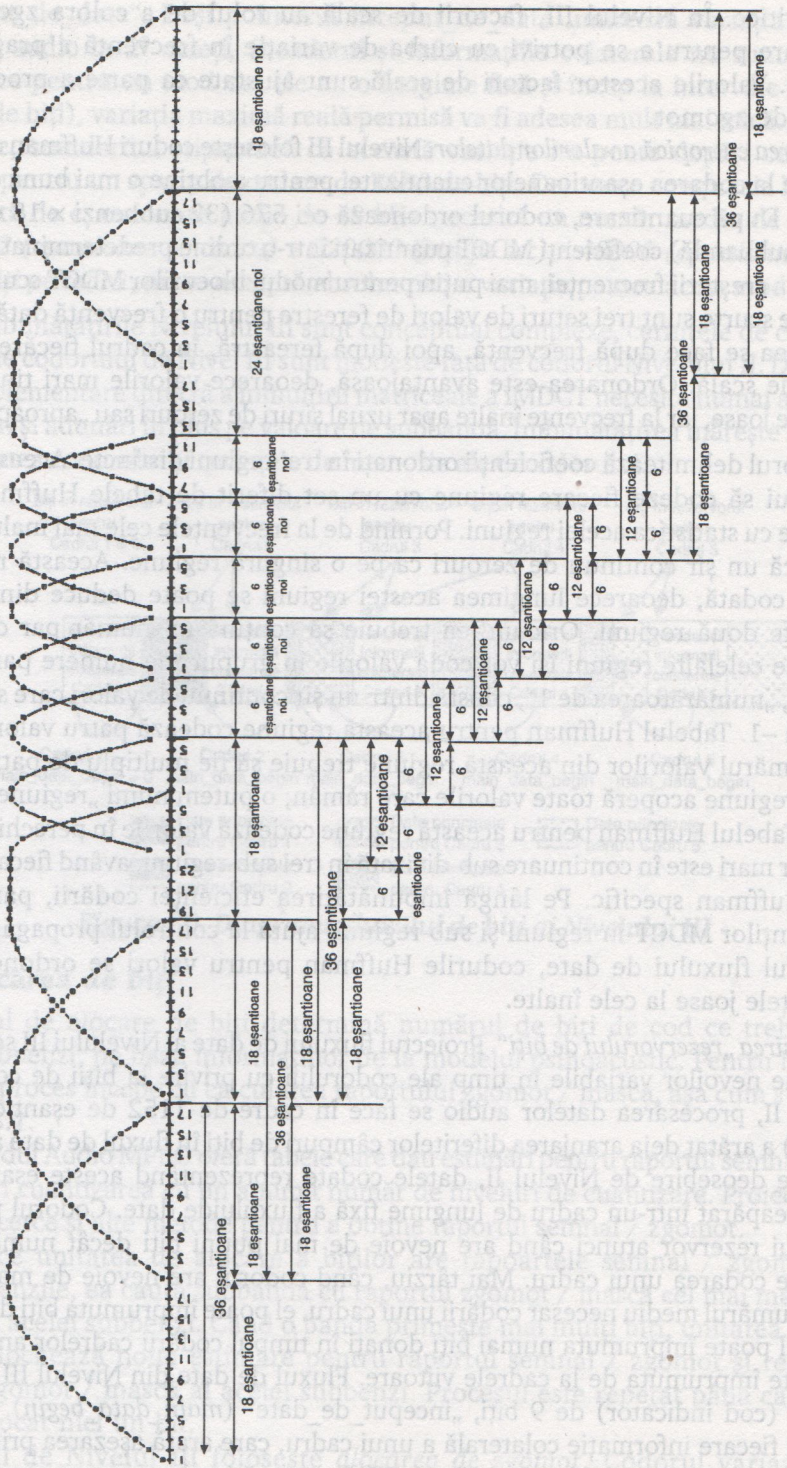


Figura 29. Aranjarea întrepătrunderii ferestrelor MDCT

benzi critice. În Nivelul III, factorii de scală au rolul de a colora zgomotul de cuantizare pentru a se potrivi cu curba de variație în frecvență a pragurilor de mascare. Valorile acestor factori de scală sunt ajustate ca parte a procesului de alocare de zgomot.

• *Codarea entropică a valorilor datelor*. Nivelul III folosește coduri Huffman de lungimi variabile la codarea eșantioanelor cuantizate, pentru a obține o mai bună compresie de date. După cuantizare, codorul ordonează cei 576 (32 subbenzi × 18 coeficienți MDCT/subbandă) coeficienți MDCT cuantizați într-o ordine predeterminată. Ordinea este cea a creșterii frecvenței, mai puțin pentru modul blocurilor MDCT scurte. Pentru blocurile scurte sunt trei seturi de valori de ferestre pentru o frecvență dată, așa încât ordonarea se face după frecvență, apoi după fereastră, în cadrul fiecărei benzi de factori de scală. Ordonarea este avantajoasă, deoarece valorile mari tind să fie la frecvențe joase, iar la frecvențe înalte apar uzual șiruri de zerouri sau „aproape zerouri".

Codorul delimitează coeficienții ordonați în trei regiuni distincte. Aceasta permite codorului să codeze fiecare regiune cu un set diferit de tabele Huffman special acordate cu statistica acelei regiuni. Pornind de la frecvențele cele mai înalte, codorul identifică un șir continuu de zerouri ca pe o singură regiune. Această regiune nu trebuie codată, deoarece lungimea acestei regiuni se poate deduce din lungimile celorlalte două regiuni. Oricum, ea trebuie să conțină un număr par de zerouri, deoarece celelalte regiuni își vor coda valorile în grupuri de numere pare. A doua regiune, „numărătoarea de 1", constă dintr-un șir continuu de valori care sunt numai 1, 0 sau –1. Tabelul Huffman pentru această regiune codează patru valori deodată, deci numărul valorilor din această regiune trebuie să fie multiplu de patru. În fine, a treia regiune acoperă toate valorile care rămân; o putem numi „regiunea valorilor mari". Tabelul Huffman pentru această regiune codează valorile în perechi. Regiunea valorilor mari este în continuare sub-divizată în trei sub-regiuni, având fiecare propriul tabel Huffman specific. Pe lângă îmbunătățirea eficienței codării, partiționarea coeficienților MDCT în regiuni și sub-regiuni, ajută la controlul propagării erorilor. În cadrul fluxului de date, codurile Huffman pentru valori se ordonează de la frecvențele joase la cele înalte.

• *Folosirea „rezervorului de biți"*. Proiectul fluxului de date al Nivelului III se potrivește mai bine nevoilor variabile în timp ale codorului cu privire la biții de cod. Ca și la Nivelul II, procesarea datelor audio se face în cadre de 1152 de eșantioane. Figura 27(c) a arătat deja aranjarea diferitelor câmpuri de biți în fluxul de date al Nivelului III. Spre deosebire de Nivelul II, datele codate reprezentând aceste eșantioane nu încap neapărat într-un cadru de lungime fixă a fluxuluide date. Codorul poate dona biți unui rezervor atunci când are nevoie de mai puțini biți decât numărul mediu cerut de codarea unui cadru. Mai târziu, când codorul are nevoie de mai mulți biți decât numărul mediu necesar codării unui cadru, el poate împrumuta biți din rezervor. Codorul poate împrumuta numai biți donați în timpul codării cadrelor anterioare; el nu poate împrumuta de la cadrele viitoare. Fluxul de date din Nivelul III include un pointer (cod indicator) de 9 biți, „inceput_de_date" (*main_data_begin*), transmis o dată cu fiecare informație colaterală a unui cadru, care arată așezarea primului octet de date audio pentru acel cadru. Figura 30 ilustrează implementarea rezervorului de biți în cadrul unei structuri de cadru de lungime fixă, cu ajutorul pointerului

„inceput_de_date". Deşi pointerul inceput_de_date limitează variaţia maximă a datelor audio la 2^9 octeţi, (headerul şi informaţiile colaterale nu sunt numărate, deoarece pentru un mod dat ele au o lungime fixă şi încep la intervale regulate în fluxul de biţi), variaţia maximă reală permisă va fi adesea mult mai mică. Din raţiuni practice, standardul stipulează că această variaţie nu poate depăşi ceea ce ar fi posibil pentru un spaţiu de codare de 7680 de biţi. Deoarece compresia la 320 kbps cu o rată de eşantionare audio de 48 kHz necesită un număr mediu de biţi de cod de *1152 (eşantioane pe cadru) × 320000 (biţi pe sec) / 48000 (eşantioane pe sec) = 7680 biţi pe cadru*, nu se acceptă absolut nici o variaţie pentru acest mod de codare.

Deşi îmbunătăţirile Nivelului III sunt conceptual complexe, cerinţele de calcul suplimentare ale codorului de Nivel III sunt modeste faţă de codorul Nivelului II. De exemplu, chiar o implementare directă a înmulţirii matriceale a IMDCT necesită numai aproximativ 19 înmulţiri şi adunări în plus pe valoare de subbandă. Îmbunătăţirea măreşte în principal complexitatea codorului şi cerinţele de memorie ale decodorului.

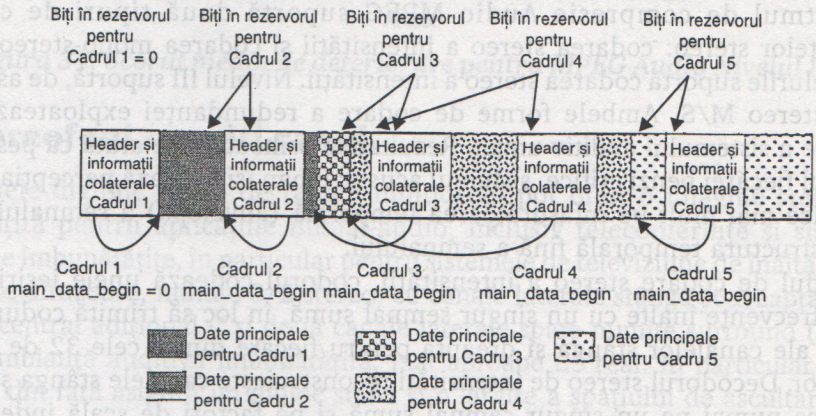


Figura 30. Diagrama fluxului de biţi ai Nivelului III

6.6. *Alocarea de biţi*

Procesul de alocare de biţi determină numărul de biţi de cod ce trebuie alocat fiecărei subbenzi, pe baza informaţiilor de la modelul psihoacustic. Pentru Nivelurile I şi II, acest proces începe cu calcularea raportului zgomot / mască, aşa cum s-a arătat în capitolul 3.1.

Standardul Audio MPEG oferă tabele care dau estimări pentru raportul semnal / zgomot rezultat din cuantizarea cu un anumit număr de niveluri de cuantizare. Proiectanţii sunt liberi să încerce şi alte metode pentru a obţine raportul semnal / zgomot.

Odată ce unitatea de alocare a biţilor are rapoartele semnal / zgomot pentru toate subbenzile, ea caută subbanda cu raportul zgomot / mască cel mai mare şi alocă biţi de cod acelei subbenzi. Când o bandă primeşte mai mulţi biţi, unitatea de alocare a biţilor blochează noua estimare pentru raportul semnal / zgomot şi recalculează raportul zgomot / mască al acelei subbenzi. Procesul este repetat până când nu mai poate fi alocat nici un bit.

Codorul de Nivelul III foloseşte *alocarea de zgomot*. Codorul variază iterativ cuantizatorii într-o manieră ordonată, cuantizează valorile spectrale, numără biţii

de cod Huffman necesari pentru codarea datelor audio şi calculează zgomotul rezultat. Dacă după cuantizare sunt încă subbenzi de factori de scală cu distorsiuni mai mari decât cele permise, codorul amplifică valorile din acele benzi de factori de scală şi micşorează efectiv treptele de cuantizare pentru acele benzi. După aceasta procesul se repetă. Procesul se opreşte dacă una dintre următoarele condiţii este adevărată:

- Nici una dintre benzile de factori de scală nu are distorsiuni mai mari decât cele permise.
- Următoarea iteraţie ar face ca amplificarea oricăreia dintre benzi să depăşească valoarea maximă admisă.
- Următoarea iteraţie ar face ca toate benzile de factori de scală să fie amplificate.

Codoarele în timp real pot să includă şi o condiţie de limită de timp pentru acest proces.

6.7. *Codarea redundanţelor stereo*

Algoritmul de compresie Audio MPEG suportă două tipuri de codare a redundanţelor stereo: codarea stereo a intensităţii şi codarea mono-stereo (M/S).[8] Toate nivelurile suportă codarea stereo a intensităţii. Nivelul III suportă, de asemenea, codarea stereo M/S. Ambele forme de codare a redundanţei exploatează o altă proprietate a sistemului auditiv uman. Rezultatele psihoacustice arată că peste 2 kHz şi în cadrul fiecărei benzi critice, sistemul acustic uman îşi bazează percepţia imaginii stereofonice mai mult pe înfăşurătoarea temporală (anvelopa) a semnalului audio decât pe structura temporală fină a semnalului.

În modul de codare stereo a intensităţii, codorul codează unele ieşiri de subbenzi de frecvenţe înalte cu un singur semnal sumă, în loc să trimită codurile independente ale canalelor stânga şi dreapta pentru fiecare dintre cele 32 de ieşiri ale subbenzilor. Decodorul stereo de intensităţi reconstruieşte canalele stânga şi dreapta bazându-se numai pe un singur semnal sumă şi pe factori de scală independenţi pentru stânga şi dreapta. La codarea stereo a intensităţii, forma spectrală a canalelor stânga şi dreapta este aceeaşi în cadrul fiecărei subbenzi de intensităţi codate, dar amplitudinea diferă.

Codoarele pentru modul stereo M/S codează semnalele canalelor stânga şi dreapta în unele benzi de frecvenţă ca mijlocul (stânga plus dreapta) şi lateralele (stânga minus dreapta). În acest mod, codorul foloseşte valori speciale de prag în vederea compresiei canalului lateralelor.

6.8. *Aprecieri subiective*

Procesul de standardizare ISO / MPEG Faza 1 a inclus teste subiective extinse (aşa numite *stimul triplu / referinţă ascunsă / dublu necunoscut*; vezi capitolul 3.3) efectuate de Swedish Broadcasting Corporation, Stockholm şi Universitatea din Hanovra (Germania) în cooperare cu Norddeutscher Rundfunk, Hanover) şi evaluări obiective ale parametrilor cum ar fi complexitatea şi întârzierea globală. Numărul total al ascultătorilor (subiecţi experţi) a fost de aproximativ 60, s-au folosit aproximativ zece secvenţe de test iar sesiunile se desfăşurau cu semnal stereo, atât cu difuzoare cât şi cu căşti. Detalii sunt date în [53] şi [54]. Ca exemplu, arătăm mai jos scorul de deteriorare rezultat pentru Nivelul II. Aceste valori sunt arătate ca bare cu trei linii în vârf. Cea din mijloc este valoarea medie,

celelalte două reprezentând media plus/minus corespunzătoare unui nivel de încredere de 95%. Ar trebui notat că s-au ales elemente critice de test astfel încât să se poată evalua codoarele în condițiile cele mai dezavantajoase, *nu* performanțele medii. Mai trebuie subliniat că aceste rezultate reflectă nivelul anului 1991; de atunci au fost aduse multe îmbunătățiri.

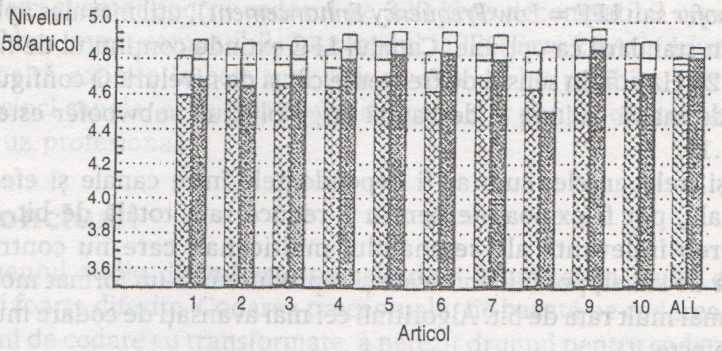


Figura 31. Scorul mediu de deteriorare pentru MPEG Audio Nivelul II[54]

7. Stereofonia multi-canal

Sistemele digitale audio de redare multicanal pot oferi o imagine stereofonică îmbunătățită pentru aplicațiile numai-audio, inclusiv teleconferințe și sisteme de televiziune îmbunătățite, în particular pentru sistemele de televiziune de înaltă definiție. Sisteme de difuzoare, numite 3/2 stereo, cu canale laterale stânga și dreapta (L și R), un canal central adițional C și două canale laterale spate numite surround (Ls și Rs), oferă o ambianță auditivă îmbunătățită, mai aproape de real. În particular, cele trei difuzoare din față asigură un mijloc stabil și o lărgire a spațiului de ascultare, atât în aplicațiile de redare audio cât și audio-vizuale, unde distorsiunile direcționale de reprezentare, adică diferențe unghiulare între imagine și sunet, pot juca un rol semnificativ. Un astfel de *format 3/2 stereo* este recomandat de ITU-R, EBU și SMPTE. O pereche suplimentară de difuzoare surround pot fi folositoare pentru situații de vizionare colectivă, de exemplu HDTV-cinema.

În Faza 2 a ISO/MPEG s-a lucrat pe partea de codare audio pentru standardizarea codării audio multicanal surround. În completare la Audio MPEG Faza 1, standardul s-a extins spre frecvențe de eșantionare mai mici și rate de bit mai mici.[55] Sistemele audio digitale multicanal suportă reprezentări *p/q*, cu *p* canale în față și *q* canale în spate. Combinațiile tipice de canale includ:

- 1 canal — modul 1/0 — centru (mono)
- 2 canale — stereo 2/0 — stânga, dreapta (stereo)
- 3 canale — stereo 3/0 — stânga, dreapta, centru
- 4 canale — stereo 3/1 — stânga, dreapta, centru, surround
- 5 canale — stereo 3/2 — stânga, dreapta, centru, surround stânga, surround dreapta

Compatibilitatea inversă de la cinci canale stereo 3/2 la mai puține canale cu reprezentare stereofonică mai redusă este asigurată pentru a răspunde nevoilor utilizatorilor care au echipamente mono sau stereo simple. În plus, sistemele stereo 3/2 vor asigura posibilitatea de a transmite două programe independente stereo și/sau un număr de comentarii sau canale multilingve. Un *canal de frecvențe joase îmbunătățit* (*subwoofer* sau LFE = *Low Frequency Enhancement*), util mai ales aplicațiilor HDTV, este inclus în standard ca opțiune. Canalul LFE extinde conținutul de frecvențe joase dintre 15 și 120 Hz atât în sensul de frecvențe cât și de niveluri. O configurație 3/2 cu cinci canale de înaltă calitate și de bandă largă plus un subwoofer este adesea numit un sistem 5.1.

Redundanțele și irelevanțele, cum ar fi dependențele între canale și efectele de mascare între canale, pot fi exploatate pentru a reduce rata totală de bit. În plus, componentele stereo irelevante ale semnalului multicanal, care nu contribuie la localizarea surselor de sunet, pot fi identificate și reproduse într-un format monofonic, pentru a reduce și mai mult rata de bit. Algoritmii cei mai avansați de codare multicanal folosesc astfel de efecte.

MPEG Audio Faza 2 include două standarde de codare audio multicanal, dintre care unul este compatibil în ambele sensuri (direct și invers) cu MPEG Faza 1.[55,56] *Compatibilitatea directă* înseamnă că decodorul multicanal poate decoda corect semnale stereo sau mono MPEG Faza 1, *compatibilitatea inversă* înseamnă că un decodor stereo MPEG Faza 1 poate livra corect un semnal stereo (față stânga plus față dreapta) când decodează un flux de informații multicanal MPEG Faza 2, adică decodorul va reproduce o mixare cu sens a celor cinci canale inițiale.

Cu compatibilitatea inversă este posibil să introducem canale multiple audio în orice moment, într-o manieră lentă, fără să facem inutile echipamentele stereo existente cu două canale. Un exemplu important este sistemul european Digital Audio Broadcasting, care va cere decodoare audio stereo MPEG-1 în prima generație, dar va putea să ofere audio multicanal mai târziu.

Partea a șaptea a standardului MPEG-2 (ISO/MPEG 13818-7) descrie aplicații care nu mai necesită compatibilitate inversă cu formatul stereo existent MPEG Faza 1. Standardul AAC (*Advanced Audio Coding* = codare audio avansată) MPEG Faza 2 folosește bancuri de filtrare de înaltă rezoluție, tehnici de predicție și codare Huffman. Standardul definește modulele de bază incluse pentru procesare opțională, transformarea din timp în frecvență (banc de filtrare), modelarea psihoacustică, predicția, cuantizarea și codarea, codarea fără zgomot și formatarea fluxului de date. Pentru a servi diferitelor necesități, standardul oferă trei profile, profilul principal, profilul de complexitate redusă și profilul cu rata de eșantionare scalabilă. De exemplu, în profilul de bază, un banc de filtrare este o transformare cosinus discretă modificată a unui bloc de 2048 sau 256; în cazul blocului de lungime mare forma ferestrei poate varia dinamic ca o funcție de semnal. Se oferă un instrument de modelare temporală a zgomotului pentru a controla dependența de timp a zgomotului de cuantizare; predicția în domeniul timp cu o predicție liniară recursivă adaptivă de ordinul 2, reduce rata de bit pentru codarea secvențelor de eșantioane ale unei subbenzi date; sunt aplicate cuantizări neuniforme și iterative și codare fără zgomot. Profilul de complexitate redusă nu folosește modelarea temporală a zgomotului și predicția în

domeniul timp, iar profilul cu rata de eşantionare scalabilă conţine un modul suplimentar de preprocesare, care permite rate de eşantionare de 6, 12, 18 şi 24 kHz.

Teste subiective oficiale s-au realizat pentru a compara codoarele MPEG Faza 2 AAC operând la 256, respectiv 320 kbps şi un codor invers compatibil MPEG Faza 2 Nivelul II, operând la 640 kbps. Toate codoarele au arătat performanţe foarte bune, cu un uşor avantaj pentru codorul pe 320 kbps necompatibil invers, faţă de codorul de 640 kbps invers compatibil. Standardul MPEG Faza 2 AAC (*Advanced Audio Coding*) oferă înaltă calitate la cele mai reduse rate de bit posibile, între 320 şi 384 kbps pentru cinci canale, el va găsi multe aplicaţii atât pentru electronica de consum cât şi pentru uz profesional.

8. Concluzii

Domeniul audio digital este pe cale de a fi aplicat în multe domenii diferite cu aplicaţii foarte diferite. Codarea preceptuală, fie bazată pe codarea subbenzilor sau pe algoritmi de codare cu transformate, a netezit drumul pentru codarea de înaltă calitate la rate de bit reduse. Codarea MPEG Faza 1 cu cele trei niveluri ale sale are potenţialul de a servi ca schemă universală de codare pentru aplicaţii cu cerinţe total diferite în domeniile electronicii de consum, a procesării audio profesionale, telecomunicaţii şi radiodifuziune. Algoritmul audio Faza 1 nu este un standard îngheţat, de vreme ce codorul său, în particular modelul psihoacustic, nu este un element normativ al standardului. Modele mai bune şi mai simple vor apărea pentru a creşte şi mai mult calitatea subiectivă a codării.

Activităţile MPEG Faza 2 au avut ca rezultat un standard care include semnale audio digitale multicanal. În modul său care nu permite compatibilitatea inversă, semnalul pentru cinci canale poate fi transmis cu o rată medie de 384 kbps sau chiar mai puţin.

Activităţile viitoare ale grupului MPEG se îndreaptă către propuneri pentru domeniul larg al aplicaţiilor multimedia. Aşteptatul standard MPEG Faza 4 va oferi rate de compresie mai mari şi va reuni toată gama de audio, de la codarea audio de mare fidelitate şi codarea vorbirii, până la vorbirea sintetică şi audio sintetic. El va oferi de asemenea şi alte funcţii, ca modificarea scalei de timp, controlul tonurilor, acces la baze de date, posibilităţi de scalare, ceea ce va permite să se extragă din fluxul de date transmis un subset suficient pentru a genera semnale audio cu lărgime de bandă redusă şi/sau o calitate mai slabă, în funcţie de capacitatea canalului sau de complexitatea decodorului. MPEG Faza 4 urma să devină un standard internaţional în noiembrie 1998.

Referinţe bibliografice

1. N. S. Jayant şi P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video,* Prentice Hall, 1984.
2. N. S. Jayant, J. D. Johnston şi Y. Shoham, "Coding of wideband speech", *Speech Commun.* **11** (1992) 127–138.
3. A. Gersho, "Advances in speech and audio compression", *Proc IEEE* **82,** 6 (1994) 900-918.
4. P. Noll, "Wideband speech and audio coding", *IEEE Commun. Mag.* **31,** 11 (1993) 34–44.
5. P. Noll, "Digital audio coding for visual communications", *Proc. IEEE* **83,** 6 (1995).

6. N. S. Jayant, "High quality networking of audio-visual information", *IEEE Commun. Mag.* **31** (1993) 84–95.
7. R. G. van der Waal, A. W. J. Oomen şi F. A. Griffiths, "Performance comparison of CD, noise-shaped CD and DCC", in *Proc. 96th Audio Engineering Society Convention,* Amsterdam, preprint 3845, 1994.
8. ISO/IEC JTC1/SC29, "Information technology–Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s–IS 11172 (part 3, audio)", 1992.
9. K. Tsutsui, H. Suzuki, O. Shimoyoshi, M. Sonohara, K. Akagari şi R. M. Heddle, "Adaptive transform acoustic coding for MiniDisc", *93rd Audio Engineering Society Convention,* San Francisco, preprint 3456, 1992.
10. T. Yoshida, "The rewritable MiniDisc system", *Proc. IEEE* **82**, 10 (1994) 1492–1500.
11. A. Sugiyama *et al.*, "A new implementation of the silicon audio player based on an MPEG/Audio decoder LSI", Tech. Rep. DSP94-99 (1994-12) of the IEICE, 1994, pp. 39–45.
12. G. Plenge, "DAB–A new sound broadcasting system", EBU Review, No. 246, 1991, pp. 1–26.
13. ACATS (Advisory Committee on Advanced Television Service), "Grand alliance HDTV system specification", Version 1.0, 1994.
14. R. Hopkins, "Choosing an American Digital HDTV terrestrial broadcasting system", *Proc. IEEE* **82**, 4 (1994) 554–563.
15. E. Zwicker şi R. Feldtkeller, *Das Ohr als Nachrichtenempfänger,* S. Hirzel Verlag, Stuttgart, 1967.
16. M. R. Schroeder *et al.*, "Optimizing digital speech coders by exploiting masking properties of human ear", *J. Acoust. Soc. Am.* **66** (1979) 1647–1652.
17. J. G. Beerends şi J. A. Stemerdink, "A perceptual audio quality measure", *92nd AES Convention,* Vienna, martie 1992, preprint 3311.
18. ITU-R, Doc. 10-2/23-E, oct. 1992.
19. K. Brandenburg şi T. Sporer, "NMR and masking flag: Evaluation of quality using perceptual criteria", *11th AES Int. Conf.,* Portland 1992, pp. 169–179.
20. R. Zelinski şi P. Noll, "Adaptive Blockquantisierung von Sprachsignalen", Tech. Rep. No. 181, Heinrich-Hertz-Institut für Nachrichtentechnik, Berlin, 1975.
21. R. E. Crochiere, S. A. Webber şi J. L. Flanagan, "Digital coding of speech in subbands", *Bell System Tech. J.* (1976) 1069–1085.
22. R. Zelinski şi P. Noll, "Adaptive transform coding of speech signals", *IEEE Trans. Acoustic, Speech and Signal Proc.* **ASSP-25** (1977) 299–309.
23. R. Zelinski şi P. Noll, "Approaches to adaptive transform speech coding at low bit rates", *IEEE Trans. Acoustic, Speech and Signal Proc.* **ASSP-27**, 1 (1979) 89–95.
24. E. P. Simoncelli şi E. H. Adelson, "Subband transforms", în *Subband Image Coding,* redactor J. W. Woods, Kluwer Acad. Publ., 1991, pp. 143–192.
25. D. Esteban şi C. Galand, "Application of quadrature mirror filters to split band voice coding-schemes", *Proc. ICASSP,* 1987, pp. 191–195.
26. J. H. Rothweiler, "Polyphase quadrature filters, a new subband coding technique", *Proc. Int. Conf. ICASSP'83,* Boston, MA, 1983, pp. 1280–1283.
27. J. Princen şi A. Bradley, "Analysis/synthesis filter bank design based on time domain aliasing cancellation", *IEEE Trans. Acoust. Speech, and Signal Process,* **ASSP-34** (1986) 1153–1161.
28. B. Edler, "Coding of audio signals with overlapping block transform and adaptive window functions", *Frequenz* **43** (1989) 252–256 (în germană).

29. M. Iwadare, A. Sugiyama, F. Hazu, A. Hirano şi T. Nishitani, "A 128 kb/s Hi-Fi audio CODEC based on adaptive transform coding with adaptive block size", *IEEE J. Sel. Areas in Commun.* **10**, 1 (1992) 138–144.
30. J. D. Johnston şi K. Brandenburg, "Wideband coding–Perceptual considerations for speech and music", *Advances in Speech Signal Processing,* redactori S. Furui şi M. M. Sondhi, Dekker, 1992, pp. 109–140.
31. K. Brandenburg, "Different kinds of source coding", *Nordisk Symp. of Sound Technique,* 1991.
32. G. T. Hathaway, "A NICAM digital stereophonic encoder", în *Audiovisual Telecommunications,* redactor N. D. Nightingale, Chapman & Hall, 1992, pp. 71–84.
33. P. Mermelstein, "G.722, a new CCITT coding standard for digital transmission of wideband audio signals", *IEEE Commun. Mag.* (1988) 8–15.
34. N. Benvenuto, G. Bertocci, W. R. Daumer şi D. K. Sparrell, "The 32-kb/s ADPCM coding standard", *AT&T Tech. J.* **65**, 5 (1986) 12–22.
35. Y. F. Dehery, G. Stoll şi 'L. V. D. Kerkhof, "MUSICAM source coding for digital sound", *17th Int. Television Symp.,* Montreux, iunie 1991, pp. 612–617.
36. G. Davidson, L. Fielder şi M. Antill, "High-quality audio transform coding at 128 kbits/s", *Proc. of the ICASSP,* 1990.
37. K. Brandenburg, J. Herre, J. D. Johnston, Y. Mahieux şi E. F. Schroeder, "ASPEC: Adaptive spectral perceptual entropy coding of high quality music signals", *90th AES Convention,* Paris 1991, preprint 3011.
38. J. D. Johnston şi A. J. Ferreira, "Sum-difference stereo transform coding", *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'92),* 1992, pp. II-569–II-572.
39. H. G. Musmann, "The ISO audio coding standard", *Proc. IEEE Globecom,* dec. 1990.
40. K. Brandenburg şi G. Stoll, "The ISO/MPEG-audio codec: A generic standard for coding of high quality digital audio", *92nd AES Convention,* Viena, martie 1992, preprint no. 3336.
41. P. Chu, "Quadrature mirror filter design for an arbitrary number of equal bandwidth channels", *IEEE Trans. ASSP* **ASSP-33**, 1 (1985) 203–218.
42. D. Pan, "Digital audio compression", *Digital Tech. J.* **5**, 2 (1993).
43. K. Konstantinides, "Fast subband filtering in MPEG audio coding", *IEEE Signal Processing Lett.* (1994) 26–28.
44. V. Bhaskaran, K. Konstantinides şi R. Lee, "Real-time MPEG-1 software decoding on HP workstations", *IS&T/SPIE Symp. on Electronic Imaging: Science and Technology, Digital Video Compression Proc.* **2419** (1995).
45. J. S. Ward şi B. J. Stanier, "Fast discrete cosine transform algorithm for systolic arrays", *Electron. Lett.* **19**, 2 (1983).
46. B. G. Lee, "A new algorithm to compute the discrete cosine transform", *IEEE Trans. ASSP* **ASSP-32**, 6 (1984).
47. W.-H. Chen, C. H. Smith şi S. C. Fralick, "A fast computational algorithm for the discrete cosine transform", *IEEE Trans. on Commun.* **COM-25**, 9 (1977).
48. J. D. Johnston, "Estimation of perceptual entropy using noise masking criteria", *Proc. Int. Conf. IEEE ASSP,* 1988, pp. 2524–2527.
49. J. D. Johnston, "Transform coding of audio signals using perceptual noise criteria", *IEEE J. Selected Areas in Communications* **6** (1988) 314–323.
50. K. Brandenburg, "OCF–A new coding algorithm for high quality sound signals", *Proc. Int. Conf. IEEE ASSP,* 1987, pp. 141–144.
51. D. Wiese şi G. Stoll, "Bitrate reduction of high quality audio signals by modeling the ears' masking thresholds", *89th AES Convention,* Los Angeles, 1990, preprint 2970.

52. K. Brandenburg şi J. D. Johnston, "Second generation perceptual audio coding: The hybrid coder", *88th AES Convention,* Montreaux, 1990, preprint 2937.
53. T. Ryden, C. Grewin şi S. Bergman, "The SR Report on the MPEG audio subjective listening tests in Stockholm April/May 1991", ISO/IEC JTC1/SC29/WGII: Doc.-No. MPEG 91/010, mai 1991.
54. H. Fuchs, "Report on the MPEG/Audio subjective listening tests in Hannover", ISO/IEC JTC1/SC29/WG//: Doc.-No. MPEG 91/331, nov. 1991.
55. ISO/IEC JTC1/SC29, "Information Technology–Generic coding of moving pictures and associated audio information–IS 13818 (Part 3, Audio)", 1994.
56. G. Stoll *et al.*, "Extension of ISO/MPEG-audio layer II to multi-channel coding: The future standard for broadcasting, telecommunication, and multimedia application", *94th Audio Engineering Society Convention,* Berlin, 1993, preprint no. 3550.

Articol din „International Journal of High Speed Electronics and Systems", Vol. 8, No. 1 (1997) 119-177
© World Scientific Publishing Company

COMPRESIA SEMNALELOR VIDEO ȘI DE IMAGINE: TRECERE ÎN REVISTĂ

CHRISTINE I. PODILCHUK și ROBERT J. SAFRANEK
AT&T Bell Laboratories
Signal Processing Research Department
Murray Hill, NJ 07974, USA
E-mail: chrisp@research.att.com

Domeniul compresiei semnalelor video și de imagine înregistrează un progres fabulos în cursul ultimelor decade. Succesele în compresia imaginilor sunt datorate perfecționării și înțelegerii superioare a metodelor de codare a formelor de undă care exploatează analiza statistică a semnalelor, a metodelor perceptuale care exploatează proprietățile psiho-vizuale ale sistemului vizual uman (HVS) și a modelelor bazate pe obiecte, în special pentru ratele foarte joase de bit. Ultimii ani au adus câteva standarde de codare a imaginii – JPEG pentru compresia imaginii și H.261, MPEG – I si MPEG – II pentru compresia video. Deși am alocat o secțiune specială a acestei lucrări pentru acoperirea standardelor internaționale de codare – datorită valorii lor practice, ne-am preocupat și de o clasă largă de tehnologii de codare nestandard, pentru viitoarea lor importanță potențială, cât și pentru întregirea abordării. Codarea semnalelor video cu rată de bit foarte joasă rămâne o problemă provocatoare întrucât presupune înțelegerea sistemului vizual uman pentru compresia optimă perceptual.
Gama largă de aplicații și rate de bit, de la video-telefonie la rate joase de 9.6 kbps și până la HDTV (High Definition TV) la 20 Mbps sau chiar mai mult, au acționat ca un catalizator pentru generarea unor idei noi în atacarea diferitelor provocări caracterizate de aplicațiile particulare. Domeniul compresiei de imagine va rămâne o arie de mare interes pentru cercetare, focalizându-ne acum pe combinarea codării surselor cu codarea canalelor și pe rețelele multimedia.

1. Introducere

Acest articol prezintă o trecere in revistă a compresiei semnalelor de imagine și video. Pentru cititorul care dorește să obțină mai multe detalii asupra oricăruia din subiectele abordate aici, am inclus o listă extinsă de referințe, care, fără a putea pretinde că este exhaustivă, reprezintă o eșantionare a câtorva arii majore de cercetare în compresia imaginii, de-a lungul vremii. Articole excelente au fost scrise asupra diferitelor aspecte ale compresiei imaginii.[164,53,120,61,114,118,10,62,5,84]

Compresia semnalelor video și de imagine este o componentă critică pentru orice aplicație care necesită transmisie sau înmagazinare de date legate de imagini. Principalul obiectiv pentru compresia video și de imagine, ca și pentru alte tipuri de compresii de semnale, este de a comprima datele pentru un canal de lărgime de bandă specificată sau de a răspunde unor necesități de stocare de date, menținând pentru semnal cea

mai înaltă calitate posibilă. Pentru multe aplicații, cea mai bună evaluare a calității semnalului se bazează pe percepția umană. Pentru transmisiile video, cercetătorii au luat în considerație problemele ridicate de căile de transmisie începând cu cele necesare pentru televiziunea de înaltă definiție (HDTV) de 20 Mbps și chiar mai mult, până la video transmis prin linii telefonice de 9,6 kbps și chiar mai puțin. O asemenea gamă largă de cerințe a compresiei semnalelor cât și exigențele fiecărei aplicații în parte au dat naștere la diferite idei referitoare la modul efectiv de creștere a eficienței compresiei datelor de imagine. Aplicația particulară generează alte constrângeri asupra proiectării codorului ca, de exemplu, întârzierea minimă pentru teleconferințe sau alte sisteme bidirecționale de comunicații, sau complexitatea rezonabil de joasă, în conexiune cu costul la utilizator. O dată cu permanenta creștere a posibilităților rețelelor și a capacității de interconectare, împreună cu dezvoltarea multimedia atât în locuințe cât și în birouri, avem nevoie de a reexamina metodele curente de compresie și ideile fundamentale care stau la baza lor și avem nevoie să inventăm noi tehnici capabile să răspundă cerințelor de viitor.

Capitolul 2 este un sumar al formatelor de imagine și a ratelor de bit – țintă, în aplicațiile de codare. Partea următoare a articolului, capitolele 3 – 5 acoperă subiectul metodologiilor de bază folosite pentru compresia datelor de imagine. O parte a lucrării a fost motivată de tehnici similare aplicate la compresia vorbirii, extinse la două dimensiuni pentru imagini statice și, uneori, la trei dimensiuni pentru video. Metodologiile de bază în compresie se împart în două categorii: metode care exploatează proprietățile statistice ale datelor de imagine și metode care se bazează pe modele. Metodele bazate pe modele pornesc ori de la modele vizuale care caracterizează redundanțele perceptuale, ori de la modele imagine pentru obiecte în spațiu și timp. Multe din codoare folosesc atât metodele tradiționale, bazate pe forma de undă, cât și unele metodologii bazate pe modele.

Capitolul 3 descrie metode de compresie a imaginii, care exploatează statistica semnalelor. Aceste tehnici pot fi destul de generale pentru a fi aplicate la mai multe tipuri diferite de semnale. Metodele au fost organizate pe categorii, deși există o oarecare suprapunere prin faptul că anumite metode pot fi încadrate în mai multe categorii. Subcapitolul 3.1 descrie metodele predictive care se fundamentează pe corelații atât spațiale cât și temporale. Metodele predictive includ modulația diferențială în pulsuri (DPCM = *Differential Pulse Code Modulation*), care poate fi aplicată spațial și temporal și compensarea mișcării (MC = *Motion Compensation*) care este aplicată temporal. Subcapitolul 3.2 descrie metodele bazate pe prelucrări tip bloc pentru compresia imaginii, care includ transformări bloc precum transformata cosinus discretă (DCT = *Discrete Cosine Transform*) ca și cuantizarea vectorială (VQ = *Vector Quantization*). Subcapitolul 3.3 acoperă sistemele multirezoluție pentru compresia imaginii. Aici este inclus și subiectul îndeobște cunoscut în literatura tehnică sub numele de codarea subbenzilor. Analiza semnalelor bazată pe descompunerea în unde mărginite (*wavelet*) este, de asemenea, tratată aici. Teoria undelor mărginite a deschis noi posibilități în acest domeniu, prin oferirea unui cadru de lucru elegant pentru proiectarea filtrelor pentru analiza / sinteza multirezoluție a semnalelor. Deși acest capitol ar putea să includă și descompunerea în frecvențe, cum ar fi DCT, am ales să tratăm transformata DCT separat în subcapitolul 3.2, deoarece ea este un element atât de critic al tuturor standardelor curente de compresie de imagine și video și pentru a putea separa metodele care procesează blocuri independente, cum este DCT, de metodele care în general nu

procesează blocuri independente, cum sunt descompunerea bazată pe subbenzi și unde mărginite (*wavelet*).[a]

Capitolul 4 descrie tehnicile de compresie a imaginii, care sunt mai puțin convenționale decât tehnicile bazate pe formele de undă și sunt fie explicit fie implicit bazate pe modele. Subcapitolul 4.1 introduce compresia imaginii bazată pe modele sau aspecte psihovizuale ale sistemului vizual uman. În codarea tradițională a formelor de undă, obiectivul este de a minimiza eroarea medie pătratică sau de a maximiza raportul semnal / zgomot (SNR). Deși SNR este folosit pe scară largă pentru a măsura performanțele de codare, este bine stabilit că SNR nu oferă o măsură de încredere pentru aprecierea calității imaginii. Acest subcapitol explorează folosirea unor modele explicite pentru alocarea optimă de bit în sens perceptual. Subcapitolul 4.2 se ocupă de tehnicile de compresie ale imaginii, bazate pe modelarea obiectelor din imagine. Metodele descrise aici vor cuprinde plaja de la modelele bine definite ale cadrelor special proiectate pentru scenele cu bust (*head-and-shoulder* = cap și umeri) până la tehnicile mai flexibile, care încearcă să păstreze diferite caracteristici ale semnalelor, cum ar fi marginile (contururi) sau zonele largi, de semnal constant, ale imaginii.

Capitolul 5 va descrie standardele curente de codare ale imaginilor fixe și ale semnalelor video. Capitolul 6 va discuta pre și postprocesarea imaginii pentru compresie iar capitolul 7 va discuta problema complexității implementării. În sfârșit, capitolul 8 descrie câteva domenii de cercetare prezente și de viitor, motivate de mulțimea de lucruri descrise în această lucrare.

2. Fundal

Pentru a comprima datele de imagine, pornim de la asumarea faptului că avem o reprezentare digitală, bazată pe eșantionarea și cuantizarea informației vizuale originale. Acest capitol furnizează un sumar al diferitelor formate de imagine, care sunt standard în mod curent pentru diferite aplicații. Definește, de asemenea, ratele – țintă pentru diferite aplicații. Mai multe detalii pot fi găsite în lucrarea de referință [119].

Standardul NTSC (*National Television System Committee*) pentru televiziunea în culori, utilizat în mod curent în Statele Unite și Japonia, are o rezoluție a imaginii de 525 linii / cadru la 59,94 câmpuri (semicadre) / secundă. Semnalul color este reprezentat prin trei componente – *Y I Q* – unde *Y* reprezintă informația de luminanță, iar *I* și *Q* sunt semnalele de crominanță diferență de culoare. Sistemul PAL (*Phase Alternation Line*) pentru televiziunea în culori, utilizat în majoritatea țărilor vest-europene, lucrează cu 625 linii / cadru si 50 câmpuri (semicadre) / secundă. Cele trei componente reprezentând semnalul de imagine sunt *Y U V*, unde *Y* este semnalul de luminanță iar *U* și *V* sunt cele două semnale de crominanță. Sistemul SECAM (*Sequential Couleur Avec Memoire*), utilizat în Franța, Orientul Mijlociu și în multe țări est europene, inclusiv Rusia este foarte asemănător cu standardul PAL. Raportul de aspect – lățimea imaginii vizibile raportată la înălțimea imaginii – este 4:3 pentru toate sistemele. Pentru televiziunea de înaltă definiție (HDTV), numărul de linii pe cadru este tipic mai mare de 1000 iar raportul de aspect este 5:3.

---

[a] Excepția acestei împărțiri pe categorii este transformarea ortogonală ciclică (LOT = *Lapped Orthogonal Transform*) care este acoperită în capitolul 3.2 deoarece ea este o extensie naturală a DCT.

Pentru a crea o reprezentare digitală compatibilă cu NTSC, PAL și SECAM, a fost creat standardul internațional CCIR 601 pentru codarea componentelor video. Acesta definește o rezoluție a imaginii de 720 pixeli / linie. Semnalele de crominanță sunt subeșantionate cu un factor 2 pe orizontală. Luminanța și două componente de crominanță definite prin acest standard sunt combinații liniare ale primarelor NTSC normalizate și cu corecție gamma, R, G, B, reprezentând componentele de roșu, verde, respectiv albastru. Componentele primare sunt normalizate fie la albul de referință NTSC fie la albul de referință PAL sau SECAM și sunt notate $\widetilde{R}$, $\widetilde{G}$ și $\widetilde{B}$. Semnalul digital de luminanță este definit ca:

$$Y_D = 219Y + 16, \tag{1}$$

unde

$$Y = 0{,}299\widetilde{R} + 0{,}587\widetilde{G} + 0{,}114\widetilde{B}. \tag{2}$$

Gama semnalului digital de luminanță este cuprinsă între 16 și 235. Semnalele diferență de culoare sunt definite ca:

$$C_B = \frac{112(\widetilde{B} - Y)}{0{,}886} + 128 \tag{3}$$

și

$$C_R = \frac{112(\widetilde{R} - Y)}{0{,}701} + 128 \tag{4}$$

unde gama celor două componente diferență de culoare este cuprinsă între 16 și 240. Valorile maxime și minime ale lui C_B corespund cu albastru, respectiv galben, iar valorile maxime și minime ale lui C_R corespund cu roșu, respectiv cyan. Toate cele trei componente sunt cuantizate cu 8 biți. Standardul de codare video H.261 a definit un format intermediar comun (CIF = *Common Intermediate Format*) cu o mărime a imaginii de 360 × 288 și o mărime a imaginii codate de 352 × 288.[b] MPEG-I a extins formatul CIF pentru a adapta diferitele formate de televiziune la formatul intermediar standard (SIF = *Standard Intermediate Format*) cu o mărime a imaginii de 360 × 240 pentru NTSC si 360 × 288 pentru PAL și SECAM. MPEG a definit și lărgimea imaginii codate ca 352 pentru a facilita codarea pe blocuri de mărime 16 × 16. Atât în SIF cât și în CIF, semnalele de crominanță sunt subeșantionate cu factorul 2, pe ambele direcții, verticală și orizontală.

Pentru aplicațiile de tipul videoconferință, ratele primare au fost definite ca 1,544 Mbps în America de Nord și Japonia și 2,048 Mbps în Europa de Vest. Rețelele de servicii digitale integrate (ISDN = *Integrated Services Digital Networks)* permit pentru transmisiile de date la multipli de 64 kbps până la ratele primare. ISDN furnizează canale B cu rate de 64 kbps și canale H0 cu rate de 384 kbps. O rată – țintă interesantă pentru televiziunea de înaltă definiție este 20 Mbps, care poate fi transmisă pe un canal de 6 MHz. Aceasta va face posibil ca acest tip de televiziune (HDTV) să fie transmisă pe un canal curent liber al NTSC.[54] Pentru video-telefonie în banda vorbirii, ratele de bit țintă sunt în gama 8 pînă la 40 kbps.

[b] 352 este ales astfel încât să ușureze codarea blocurilor de dimensiuni 16×16.

3. Tehnici de compresie bazate pe forma de undă

Modulația impulsurilor în cod (PCM = *Pulse Code Modulation*) constă în eșantionarea unui semnal analogic și cuantizarea fiecărui eșantion folosind un set finit de nivele de cuantizare. Aceasta descrie complet modul de bază al sistemului de codare. PCM a fost aplicată pentru digitizarea unui semnal de televiziune pentru prima dată în 1951.[50] Deși PCM nu este un sistem de codare eficient, vom revedea pe scurt procesul de cuantizare, deoarece este o parte importantă și integrantă a tuturor tehnicilor de compresie care vor urma.

Cuantizarea fără memorie poate fi descrisă simplu ca transformarea amplitudinii unui semnal $x(n)$ la momentul n[c] într-o amplitudine $y(n)$ aleasă dintr-un set finit de amplitudini posibile. Un cuantizator de nivel L este caracterizat de *intervalele de decizie*, I_k, și *nivelurile de refacere*, y_k, unde

$$y = y_k \text{ dacă } x \in I_k \tag{5}$$

și

$$I_k : \{x_k < x \leq x_{k+1}\} \; k = 1, 2, \ldots, L. \tag{6}$$

Un cuantizator scalar poate fi *uniform* sau *neuniform*, după cum intervalele de decizie și nivelurile de reconstrucție sunt sau nu o funcție liniară simplă de k. Cuantizatorii care sunt optimizați să minimizeze eroarea medie pătratică (mse) sunt cunoscuți sub numele de cuantizatori *Lloyd-Max*.[90,103,91] Ideea de bază este de a modela cuantizatorul pentru a avea intervale de decizie mai mici acolo unde funcția de densitate probabilistică (pdf = *probability density function*) este mare și intervale de decizie mai largi acolo unde pdf este mai mică. Acești cuantizatori neuniformi bazați pe pdf pot fi calculați iterativ.

Unele lucrări interesante legate de cuantizarea imaginii includ folosirea oscilațiilor (*dithering*) pentru atenuarea efectelor proaste ale cuantizării. O cuantizare de proastă calitate va avea drept rezultat un semnal de eroare de cuantizare care este corelat într-un grad înalt cu semnalul de intrare și produce efecte dezagreabile din punct de vedere vizual. Prin introducerea unui semnal de înaltă frecvență, determinist sau pseudo-aleator, în semnalul de intrare înainte de cuantizare, efectele de cuantizare dependente de semnal pot fi reduse. Pentru cuantizarea imaginii, aceasta este o metodă specială, efectivă, pentru a suprima zgomotul de contur, care reprezintă o schimbare abruptă în nivelul de ieșire pentru o schimbare lentă a imaginii de la intrare.[154]

Dacă semnalul de intrare este nestaționar, ceea ce este adevărat pentru date video și de imagine ca și pentru alte semnale din „lumea reală", are sens să considerăm adaptarea cuantizatorului la schimbarea caracteristicilor semnalului de intrare. Adaptarea nerecursivă necesită transmiterea unor informații suplimentare decodorului, în timp ce adaptarea recursivă se bazează numai pe istoria recentă a ieșirilor cuantizate. Multe din codoarele de imagine curente se bizuie în proiectarea lor pe câte un anume tip de cuantizare adaptivă.

Până acum am examinat cuantizarea datelor privite în mod independent. Pentru a obține o compresie înaltă, avem nevoie să exploatăm dependențele dintre pixelii de imagine, atât spațial cât și temporal.

---

[c] Timpul discret n poate fi înlocuit cu poziția spațială i, j.

3.1. *Metode predictive*

O cale de compresie a datelor de imagine este de a exploata corelațiile dintre pixeli învecinați în timp și spațiu. Codarea predictivă este ideea de bază a estimării datei prezente pe baza datelor codate anterior. În loc de a coda direct semnalul, eroarea de predicție este cea codată și transmisă. Pentru datele de imagine fixă sunt considerate corelațiile spațiale, în timp ce pentru video (imagini în mișcare) putem profita de corelațiile atât temporale cât și spațiale. Pentru video, predictoarele se pot baza numai pe pixelii imaginii prezente sau pe cadrul video actual și cel anterior. Acestea sunt cunoscute ca predicții *intra-cadru*, respectiv *inter-cadre*. Predictorii pot fi atât liniari sau neliniari, cât și ficși sau adaptivi. Un bun exemplu de predictor adaptiv este o tehnică prin care se comută între predicția liniară inter-cadru și cea intra-cadru, funcție de caracteristicile spațio-temporale ale semnalului video care este codat.

DPCM

O cale de a extinde modulația impulsurilor în cod (PCM) la codarea datelor corelate este cunoscută ca modulație diferențială a impulsurilor în cod (DPCM). Se face o distincție între codarea predictivă, unde eroarea de predicție este codată cu numai 1 bit sau 2 nivele (cunoscută ca modulație Delta, DM) și codarea în care eroarea de predicție este codată cu mai mult de un bit (la care ne vom referi ca DPCM). Deși DM a fost folosită pentru a coda vorbirea, nu a fost aplicată extensiv la codarea datelor video. Acesta se datorează în parte și ratelor mari de eșantionare, necesare pentru DM. Predicția în DPCM poate fi unidimensională, atunci când un pixel este prezis pe baza pixelilor anteriori de pe aceeași linie, bidimensională atunci când predicția se face pe pixelii din aceeași linie și din liniile precedente și tridimensională atunci când predicția se face pe baza pixelilor din același cadru și din cadrele precedente. Se pot formula două forme de DPCM: *cu buclă deschisă*, unde predicția se bazează pe eșantioanele precedente și se exprimă ca

$$\hat{x}(n) = \sum_{m=1}^{M} a_m x(n-m), \tag{7}$$

și predicția *cu buclă închisă*, unde predicția se bazează pe valorile cuantizate ale eșantioanelor precedente și se exprimă ca

$$\hat{x}(n) = \sum_{m=1}^{M} a_m \tilde{x}(n-m). \tag{8}$$

Aici, a_i corespunde coeficienților de predicție iar $\hat{x}(n)$ este valoarea prezisă a lui $x(n)$ și $\tilde{x}(n)$ este valoare cuantizată a lui $x(n)$.[d] Altă terminologie utilizată curent pentru evaluarea sistemelor DPCM este *eroarea de predicție*, care este definită ca

$$d(n) = x(n) - \hat{x}(n), \tag{9}$$

și *câștigul de predicție*, care este exprimat ca

$$G_p = \frac{\sigma_x^2}{\sigma_d^2}. \tag{10}$$

[d] Aici, DPCM este descrisă pentru un semnal unidimensional $x(n)$ dar poate fi generalizată la N dimensiuni; pentru datele video, unde vrem să profităm de corelațiile atât temporale cât și spațiale, semnalul unidimensional $x(n)$ este înlocuit cu $x(i,j,t)$ unde i, j descriu poziția spațială, iar t reprezintă componenta temporală.

Câştigul de predicţie reprezintă îmbunătăţirea raportului semnal / zgomot prin trecerea de la PCM la DPCM.

DPCM este aplicată curent la datele de imagine, folosind un predictor liniar bidimensional.[63,53,112,120] Chiar dacă un predictor bidimensional nu aduce o îmbunătăţire semnificativă a raportului semnal / zgomot, faţă de un predictor unidimensional, predictorul bidimensional conduce la o mai bună redare a marginilor obiectelor (contururilor) decât un predictor unidirecţional. Predicţia adaptivă liniară este, de asemenea, foarte utilă pentru datele de imagine. De obicei, corelaţia direcţională este estimată din pixelii transmişi anterior şi un predictor se alege pe baza direcţiei corelaţiei maxime.[51] Predictori intercadre, mai sofisticaţi, care se bazează pe clasificarea pixelilor în zone staţionare sau nestaţionare, au fost de asemenea examinaţi.[56] O formă comună de predicţie intercadre, cunoscută ca *umplerea condiţionată*[110] (*conditional replenishment*) actualizează numai pixelii care s-au schimbat semnificativ faţă de valorea pixelilor de pe aceeaşi poziţie din cadrul precedent. Aceasta se poate descrie ca o predicţie de ordinul unu urmată de o cuantizare a erorii de zonă nulă.

Compensarea mişcării

Deşi am ales să prezentăm compensarea mişcării în capitolul de tehnici de predicţie, ea poate fi, de asemenea, descrisă în capitolul despre tehnicile bazate pe model, de vreme ce se referă la un model simplu de estimare a mişcării. Pentru compresia video, am dori să exploatăm corelaţia puternică între cadre. Un set de cadre consecutive într-o secvenţă video poate fi gândită ca un set de obiecte deplasate puţin de la un cadru la celălalt. Cadrul curent poate fi prezis destul de bine din cadrul precedent, cu vectorii de deplasare potriviţi, pentru obiectele din scenă. Articole excelente despre problema generală a estimării mişcării au fost deja scrise.[2] În scenele reale, mişcarea este complexă iat obiectele nu sunt rigide. Cele mai multe dintre tehnicile de estimare a mişcării se bazează pe un model foarte simplu. Tehnicile presupun că mişcarea este o translaţie şi se face într-un plan paralel cu planul camerei de filmat. Aceasta exclude rotaţia şi zoomul. Metodele presupun, de asemenea, că lumina este spaţial şi temporal constantă, aşa că se pot face aprecieri pe baza intensităţii pixelilor dintre cadre. Multe metode ignoră, de asemenea, mascarea unor obiecte şi descoperirea fundalului. Cu aceste presupuneri, estimarea mişcării de translaţie între cadre poate fi uşor calculată. Două tehnici curent folosite pentru predicţia mişcării compensate în codarea video sunt metodele bazate pe blocuri şi metodele recursive. Metodele bazate pe blocuri presupun că mişcarea este constantă în interiorul unui bloc mic de date de imagine. Cu cât blocul este mai mic, cu atât modelul este mai valabil, dar cu preţul creşterii surplusului pentru transmisia informaţiei vectorilor de mişcare către decodor. Un vector de deplasare $d_k = (i_k, j_k)$ este calculat pentru fiecare bloc b_k, centrat pe poziţia $k = (i,j)$, din cadrul t, prin minimizarea unei funcţii de cost care măsoară similitudinea dintre b_k din cadrul t şi blocurile de imagine din cadrul de la $t - t_0$,

$$\min_{d_k}\{C(d_k, k, t) = F(b(k,t), b(k - d_k, t - t_0))\}. \tag{11}$$

O posibilitate este de a minimiza eroarea medie pătratică (mse), adică:

$$\min_{d_k}\{\sum\sum[b(k,t) - b(i - i_k, j - j_k, t - t_0)]^2\}. \tag{12}$$

În scopul reducerii complexității de calcul și în vederea unor bune performanțe, o funcție de cost mult utilizată pentru estimarea mișcării bazate pe blocuri (MC), este diferența medie absolută (MAD = *Mean Absolute Difference*), definită ca:

$$\min_{d_k}\{\sum\sum|b(k,t)-b(i-i_k,j-j_k,t-t_0)|\}. \quad (13)$$

Această tehnică poate fi descrisă ca o predicție de ordinul unu bazată pe blocuri. În vederea găsirii celei mai bune împerecheri cu o deplasare maximă d_{max}, o tehnică exhaustivă ar cere calcularea funcției de cost precedente de $(2d_{max}+1)^2$ ori. S-au propus câteva tehnici de a reduce complexitatea căutării pentru MC, bazate pe blocuri. Ele includ o căutare 2D-logaritmică[114], o procedură de căutare în trei pași[114] și o căutare conjugată a gradientului.[114] Aceste tehnici sunt euristice și nu garantează o adaptare optimă. Compensarea mișcării bazată pe blocuri, cu acurateți de fracțiuni de pixeli, a fost studiată pentru aplicații de codare video.[49] Compensarea mișcării cu o acurateți de pixeli (întregi) constrânge vectorii de mișcare să fie deplasați pe orizontală și pe verticală cu multipli întregi ai grilei de eșantionare. Întrucât grila de eșantionare nu este legată de deplasarea reală între cadre, acuratețea de fracțiuni de pixeli poate conduce la o mai bună predicție a compensării mișcării. Acuratețea de fracțiuni de pixeli este obținută prin interpolarea imaginilor cadrelor înainte de a aplica compensarea mișcării. Rezultatele din [49] arată că pentru compensarea mișcării cu o dimensiune a blocurilor de 16×16, o acurateți de ¼ pixeli este de dorit pentru o calitate de emisiune radiodifuzată, în timp ce o acurateți de ½ pixeli este suficientă pentru un videofon. MC bazată pe blocuri este implementată în toate standardele curente de compresie video și este larg folosită pentru estimarea mișcării video în algoritmii de căutare.

O altă abordare pentru estimarea mișcării se bazează pe algoritmi recursivi care obțin iterativ estimări ale deplasării, bazându-se pe estimările precedente.[122,123] Estimările pot fi realizate la fiecare pixel sau la fiecare bloc de pixeli. Aceste metode exploatează relațiile care există între corelațiile spațiale și temporale ale datelor video. Metoda descrisă în [122] folosește un algoritm abrupt de coborâre pentru a minimiza funcția de cost. Cercetătorii au investigat metode de reducere a cerințelor de calcul ale metodelor recursive de estimare a mișcării. Această muncă include optimizări neliniare pentru o convergență mai rapidă[171], o strategie de căutare a asemănărilor maxime și programare dinamică,[17] și un algoritm de căutare iterativă cu arbori binari.[129] Un dezavantaj al estimării recursive a mișcării este marea complexitate de calcul la decodor și dificultățile de a urmări schimbările mișcării între cadre și discontinuitățile de mișcare din același cadru. Metodele bazate pe blocuri transmit estimările mișcării la decodor astfel încât variații ale mișcării câmpurilor pot fi bine urmărite. Oricum, atașarea unui câmp constant de mișcare fiecărui bloc de date de imagine duce la estimări de mișcare imperfecte, cel mai vizibil la contururile obiectelor în mișcare.

O altă lucrare interesantă despre compensarea mișcării pentru codarea video este [132]. Unul din punctele slabe ale compensării mișcării bazate pe prelucrări de bloc este o compensare lipsită de acurateți a marginilor obiectelor, care nu coincid cu marginile blocului. La rate de bit scăzute, eroarea de predicție poate să nu fie codată cu precizie și artefacte (semnale false) de bloc pot să apară la conturul obiectelor în mișcare. În lucrarea de referință [132], metoda originală pe bază de blocuri pentru estimarea mișcării este relaxată pentru a furniza o estimare mai precisă a câmpului în mișcare. Fiecare bloc este

forțat să aibă cel mult doi vectori de mișcare diferiți, determinați de o segmentare a câmpului de mișcare cu două constrângeri – blocurile de la marginile obiectelor în mișcare conțin cel mult două regiuni conectate și dacă unei regiuni îi este alocat un vector de mișcare al unui bloc învecinat, ea trebuie să fie conectată la acel bloc. Un câmp aleator Markov (MRF = *Markov Random Field*) este utilizat ca model *a priori* pentru câmpul de segmentare. Rezultatele codate la 1 Mbps pentru sistemul de mai sus bazat pe o prelucrare de regiuni, arată o economie de 20-30% la rata de bit, în comparație cu compensarea tradițională a mișcării, bazată pe blocuri.

Cele mai bine stăpânite și practicate sisteme pentru compensarea mișcării sunt acelea bazate pe procesarea tradițională de bloc și pe mișcarea de translație. Aceste tehnici sunt cunoscute ca fiind stabile și rezonabil de robuste pentru o largă varietate de semnale și condiții de zgomot.

3.2. *Metode bazate pe blocuri: codarea prin transformări și cuantizarea vectorială*

Acest paragraf descrie metodele bloc pentru compresia imaginilor. Și anume va fi discutată transformata cosinus discretă (DCT = *Discrete Cosine Transform*), un bloc constituent major al tuturor standardelor curente de compresie pentru imagini statice și video. Detalii asupra felului cum sunt codați coeficienții DCT pentru schemele particulare de compresie sunt furnizate în secțiunea despre standarde (capitolul 5). Altă schemă de compresie bazată pe blocuri, care va fi descrisă aici, este cuantizarea vectorială (VQ = *Vector Quantization*). Vor fi date și câteva exemple despre modul de aplicare al VQ la codarea imaginii și a semnalelor video.

Codarea prin transformări

Ideea acestui tip de codare este de a aplica un operator liniar semnalului original, în scopul obținerii unei reprezentări în care datele sunt statistic independente sau măcar necorelate. În mod ideal, transformarea liniară ar trebui să asigure maximum de compactizare energetică, o proprietate urmărită în compresia semnalelor.

O motivare directă, care arată de ce DCT este folositoare pentru compresia imaginilor, este dată de relația care există între DCT și transformata Karhunen-Loeve (KLT). Transformata Hotelling[59], sau transformata Karhunen-Loeve (KLT)[65], oferă o transformare ortogonală, dependentă de semnal, al cărei rezultat este un set de coeficienți de transformare necorelați, pentru un semnal colorat (diferit de semnalul alb). Pentru un semnal de intrare unidimensional x, definim matricea de covarianță R_{xx} astfel:

$$R_{xx} = E[xx^T]. \tag{14}$$

R_{xx} este o matrice Toeplitz simetrică și toate valorile ei principale sunt reale și nenegative (adică matricea R_{xx} este pozitiv semidefinită).[130] Transformata KLT este definită ca transformata A_{KL} care diagonalizează covarianța blocului transformat X, adică,

$$R_{XX} = A_{KL}^T R_{xx} A_{KL} = \Lambda \tag{15}$$

unde Λ este o matrice diagonală,

$$\Lambda = diag\{\lambda_0, \lambda_1, \ldots, \lambda_{M-1}\}. \tag{16}$$

Matricea A_{KL} care satisface ecuația (15) este matricea ortogonală ale cărei coloane sunt vectorii proprii ai R_{xx} și elementele din diagonala lui R_{XX} sunt valorile proprii ale lui R_{xx}, adică,

$$R_{xx}A_k = \lambda_k A_k, \tag{17}$$

unde A_k reprezintă coloana numărul k a matricei A, sau vectorul propriu cu numărul de ordine k din R_{kk}. Funcțiile de bază are transformării KLT sunt vectorii proprii ai matricei de covarianță a semnalului de intrare. Atunci când semnalul de intrare este un proces Markov de ordinul întâi sau AR(1), funcțiile de bază ale transformării KLT devin sinusoide. Prin diagonalizarea matricei de covarianță, transformarea KLT maximizează compactizarea energiei în X. Câștigul de codare al transformării, G_{TC}, este definit ca și câștigul sau reducerea erorii medii pătratice obținute prin aplicarea cuantizării scalare a coeficienților de transformare în locul codării PCM a datelor originale ale eșantioanelor[63];

$$G_{TC} = \frac{\frac{1}{M}\sum_{k=0}^{M-1}\sigma_k^2}{\left(\prod_{k=0}^{M-1}\sigma_k^2\right)^{\frac{1}{M}}}. \tag{18}$$

Transformarea KLT maximizează câștigul de codare al transformatei, dat de ecuația (18). În practică, nu se folosește transformarea KLT pentru compresia imaginii deoarece transformata este dependentă de semnal și ar trebui actualizată permanent pentru un semnal nestaționar.

Transformata cosinus discretă (DCT). O transformare liniară care este larg folosită pentru compresia datelor de imagine este DCT. Inițial, DCT a fost derivată din polinoamele Cebâșev pentru a aproxima vectorii proprii ai matricei de autocorelație a unui proces AR(1).[3] S-a arătat în [29] că pentru un semnal AR(1), cum coeficienții de corelație între eșantioane adiacente se apropie de 1, funcțiile de bază ale KLT devin funcțiile de bază ale DCT. Transformarea directă unidimensională pentru N puncte DCT (FDCT = *Forward DCT*) și inversă (IDCT = *Inverse DCT*) sunt definite ca

$$X(u) = \frac{c_u}{2}\sum_{i=0}^{N-1} x(i)\cos\left[(2i+1)\frac{u\pi}{2*N}\right], \tag{19}$$

și

$$x(i) = \sum_{u=0}^{N-1}\frac{c_u}{2}X(u)\cos\left[(2i+1)\frac{u\pi}{2*N}\right], \tag{20}$$

unde

$$c_u \equiv \begin{cases} \dfrac{1}{\sqrt{2}} & \text{pentru } m=0 \\ 1 & \text{în rest} \end{cases}$$

În mod tipic, pentru aplicațiile de codare de imagine și video, se aplică o transformare DCT bidimensională unui bloc de date de imagine, prin aplicarea unei DCT unidimensionale rândurilor și coloanelor, separat. Cele două transformate bidirecționale, DCT directă (FDCT) și inversă (IDCT), pot fi exprimate ca

$$X(u,v)=\frac{c_u}{2}\frac{c_v}{2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}x(i,j)\cos\left[(2i+1)\frac{u\pi}{2*N}\right]\cos\left[(2j+1)\frac{v\pi}{2*n}\right], \quad (21)$$

și

$$x(i,j)=\sum_{u=0}^{N-1}\frac{c_u}{2}\sum_{v=0}^{N-1}\frac{c_v}{2}X(u,v)\cos\left[(2i+1)\frac{u\pi}{2*N}\right]\cos\left[(2j+1)\frac{v\pi}{2*n}\right], \quad (22)$$

unde

$$c_u \equiv \begin{cases}\frac{1}{\sqrt{2}} & \text{pentru } m=0\\ 1 & \text{în rest}\end{cases}$$

și

$$c_v \equiv \begin{cases}\frac{1}{\sqrt{2}} & \text{pentru } n=0\\ 1 & \text{în rest}\end{cases}$$

DCT a fost folosită extensiv pentru codarea de imagine și video și este una dintre principalele componente ale standardelor curente de imagine și video (vezi capitolul 5 pentru mai multe detalii). Datorită importanței DCT pentru aplicațiile de codare a imaginii, s-a consumat multă muncă pentru a reduce complexitatea de calcul a DCT. Ca și în cazul transformatei Fourier rapide (FFT), care exploatează simetria transformatei Fourier discrete, se poate exploata simetria DCT pentru a obține versiuni mai rapide.[85] În mod curent, algoritmii cei mai rapizi au fost introduși în două metode care mapează (transformă) DCT într-o altă transformare – DFT[181] și o versiune deplasată în frecvență a DCT cunoscută ca DCT-IV[185]. În ambele cazuri, pentru o DCT N-dimensională, numărul de multiplicări este $(N/2)\log_2 N+1$ și numărul de adunări este $(3N/2)\log_2 N-N+1$. Pentru un studiu detaliat al DCT, vedeți [151].

Transformări care sunt mai puțin complexe, ca Haar, Walsh-Hadamard și transformarea oblică (*slant transform*)[52,145,146] au fost, de asemenea, aplicate la codarea imaginii dar rezultatele DCT au asigurat performanțe mult mai bune pentru datele de imagine.

Una dintre limitările DCT bazate pe blocuri, pentru codarea imaginii, este că la rate de bit reduse, codorul introduce discontinuități la granițele blocurilor, care apar ca artefacte (delimitări false) de „bloculețe". Transformarea ortogonală întrețesută (LOT = *Lapped Orthogonal Transform*) îndulcește acest efect prin suprapunerea blocurilor de date. Cu alte cuvinte, dacă dimensiunea blocului este *N*, transformarea tradițională DCT sau o abordare bazată pe blocuri transformă cele *N* eșantioane în *N* coeficienți de transformare. Pe de altă parte, LOT transformă *L* eșantioane de intrare în *N* coeficienți de transformare, unde $L > N$. De vreme ce rata de eșantionare rămâne constantă, cei *N* coeficienți de transformare ar trebui calculați pentru fiecare *N* eșantioane de intrare. Aceasta conduce la o întrepătrundere de $L - N$ eșantioane în fiecare două blocuri consecutive. Pentru mai multe detalii despre transformarea ortogonală întrețesută, transformarea întrețesută extinsă

(ELP = *Extended Lapped Transform*) şi transformarea modulată întreţesută (MLT = *Modulated Lapped Transform*) şi despre modul în care se leagă acestea de descompunerea în subbenzi, consultaţi [99]. Deşi LOT oferă o calitate mai bună decât DCT, aceasta se face cu preţul unei complexităti de calcul suplimentare şi nu suportă scheme MC pentru compresie video. Datorită acestor motive şi parţial datorită altora de ordin istoric, în ciuda proprietăţilor lor foarte bune, metodele de transformare întreţesută nu şi-au găsit încă aplicaţii comerciale largi în tehnologiile de codare video şi a imaginii. O altă lucrare recentă despre evitarea artefactelor de bloculeţe este [23], unde DCT pe bază de blocuri este realizată pe imagini segmentate şi segmentele sunt extrapolate la dimensiunea blocurilor DCT prin folosirea teoriei proiectării pe seturi convexe (POCS = *Projection Onto Convex Sets*).

DCT este fundamentul tuturor standardelor curente de codare a imaginii, atât pentru imagini statice (JPEG) cât şi pentru video (H.261, MPEG-I, MPEG-II). În particular, cel mai des folosită este DCT 8×8. O ilustrare a funcţiilor de bază ale DCT 8×8 apare în figura 1.

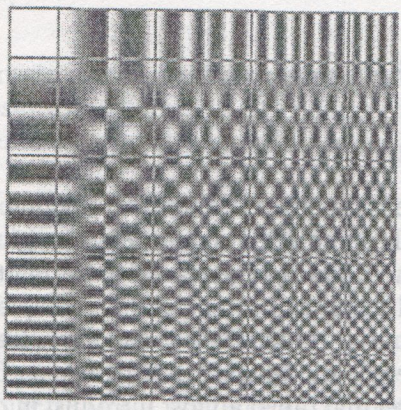


Figura 1. Ilustrarea funcţiilor de bază ale DCT 8×8

Cuantizarea vectorială

Conform teoriei ratei de distorsiuni a lui Shanon, cele mai bune performanţe de codare se obţin prin codarea vectorilor în locul scalarilor, chiar pentru surse fără memorie. Cuantizarea vectorilor a fost dezvoltată mai ales pentru aplicaţii de compresie a datelor. Ideea de cuantizare vectorială (VQ) a fost larg studiată şi aplicată extensiv la codarea imaginilor statice şi a datelor video. Pentru mai multe detalii privind VQ şi diferitele ei aplicaţii, vedeţi [46]. Primele lucrări despre folosirea VQ pentru codarea imaginilor includ [195], [45], [11]. Un articol pe larg despre VQ pentru compresia imaginilor statice apare în [118]. Ideea de bază a cuantizării vectoriale este de a transforma un vector de intrare de dimensiune k într-un set finit C de M vectori de reproducere. Fiecare vector de intrare x este mapat (transformat) într-unul din vectorii de reproducere M cuprinşi în matricea de codare C, adică

$$Q : R^k \rightarrow C. \tag{23}$$

unde $C = (y_1, y_2, \ldots, y_M)$ şi $y_i \in R^k$ pentru fiecare $i \in I = \{1,2,\ldots,M\}$. Matricea de codare C este de dimensiune M şi fiecare dintre cei M vectori de cod din C are dimensiunea k. Rata cuantizatorului vectorial este dată de

$$r = \frac{\log_2 M}{k}. \qquad (24)$$

Fiecărei VQ de M puncte îi este asociată o partiție a spațiului R^k în M regiuni sau celule, R_i, $i = 1,2, \ldots, M$, unde

$$R_i = \{x \in R^k : Q(x) = y_i\}. \qquad (25)$$

O metodă larg folosită pentru proiectarea cuantizatorilor vectoriali este agoritmul iterativ cunoscut ca Algoritmul Lloyd generalizat [91] sau, de asemenea, numit *algoritmul celor k medii.*[95] Algoritmul LBG introdus de Linde, Buzo și Gray [87] este o variantă a algoritmului Lloyd generalizat cu un algoritm special de divizare pentru generarea agendei de coduri. Algoritmul Lloyd generalizat, pentru proiectarea cuantizatorilor vectoriali, poate fi descris în următorii pași:

- Algoritmul Lloyd generalizat

1. Se inițializează matricea de codare C_0 ; $l = 1$
2. Se împart datele de lucru în R_i, folosind cea mai apropiată condiție de vecinătate:
   $R_i = \{x \in T : d(x, y_i) \le d(x, y_j)$; pentru orice $j \ne i\}$.
3. Se calculează centroizii pentru noile grupe:
   $C_{l+1} = \{cent(R_i)\}$
4. Se calculează distorsiunea medie D_{l+1} pentru C_{l+1}.

   dacă $\frac{D_l - D_{l+1}}{D_l} \le T$ atunci *stop*

   dacă $\frac{D_l - D_{l+1}}{D_l} > T$ $l = l + 1$ atunci *go to 1* (treci la pasul 1)

T reprezintă pragul care determină când trebuie să se oprească iterația. Căutarea completă VQ necesită măsurarea distorsiunilor pentru toți vectorii matricii de codare și alegerea minimului. S-a cheltuit multă muncă pentru a reduce complexitatea de calcul a VQ prin reducerea căutării. O abordare este de a impune restricții structurale matricii de codare VQ. O astfel de tehnică larg utilizată este cunoscută ca TSVQ (*Tree-Search VQ* = căutare arborescentă VQ)[43] unde căutarea se face pe etape. Această abordare minimizează complexitatea căutării VQ, cu prețul unei capacități de stocare suplimentare și a lipsei de garanție asupra găsirii reprezentării optime, așa cum este definită de metrica distorsiunilor. Codorul pentru TSVQ este structurat arborescent, iar căutarea și decizia se face la fiecare nivel al arborelui, înainte de avansarea la nivelul următor.

Cea mai directă metodă de a aplica VQ la imagini este în domeniul spațial, unde vectorii de cod reprezintă un bloc bidimensional al imaginii de intrare. Datorită nestaționarității datelor de imagine, s-au făcut încercări de a normaliza imaginea înainte de VQ. Aceasta include lucrarea [111], unde fiecare vector este normalizat la o medie zero și varianță unitate înainte de VQ, iar mediile și variațele sunt cuantizate scalar și transmise ca informație colaterală, precum și lucrarea [12] numită și VQ de formă medie și reziduu mediu, unde media eșantioanelor este scăzută din vectori înainte de VQ. O altă abordare spațială pentru a stăpâni nestaționaritatea datelor de imagine este cunoscută ca

CVQ (= *Classified VQ*)[149] unde datele de imagine sunt segmentate în diferite clase și se alocă fiecărei clase o matrice de codare. Datele sunt clasificate în regiuni plate (fără variații ale semnalului) și regiuni caracterizate de fronturi (contururi) de diferite orientări.

Cuantizarea vectorială predictivă, cunoscută ca FSVQ (= *Finite State VQ*) încearcă să exploateze la codare corelațiile intervectoriale. Ideea de bază este de a obține calitatea oferită de o matrice de codare mare, în timp ce se menține rata unei matrici de codare mult mai mici. Codorul și decodorul unui sistem FSVQ au structura unei mașini cu număr finit de stări. O matrice de codare separată C_s există pentru fiecare stare S. Funcția de tranziție a stărilor mapează (transformă) starea curentă și simbolul de canal în starea următoare. În [9], FSVQ este aplicată la codarea imaginii prin utilizarea corelațiilor geometrice și de intensitate, care există între blocurile vecine, pentru a determina funcția de tranziție a stărilor. Un cuantizator vectorial diferențial introdus în [34] constă dintr-un predictor liniar vectorial și un cuantizator vectorial al erorii de predicție. Un cuantizator vectorial predictiv pentru codarea imaginii este descris în [55].

Două metode FSVQ – SMVQ (*Side Match VQ*) și OMVQ (*Overlap Match VQ*) – exploatează marea corelație spațială a pixelilor într-o imagine și contiguitatea valorilor pixelilor de-a lungul marginilor blocurilor.[69,70] SMVQ și OMVQ încearcă să minimizeze marginile vizibile ale blocurilor create de VQ tradițională la rate de bit reduse pentru compresia imaginii. SMVQ presupune că rândurile și coloanele de pixeli într-o imagine sunt un proces Markov de ordinul I. Matricea de codare a stărilor este aleasă în așa fel încât să fie cea a cărei vectori de coduri au cea mai bună „potrivire (adaptare) a marginilor"; cu alte cuvinte, vectorii care coincid cel mai bine cu prima linie și prima coloană în blocul ce urmează a fi codat. Cele două blocuri codate, care sunt adiacente primei linii și primei coloane, determină starea. Dacă presupunem o ordonare în sensul de scanare a rastrului, blocurile codate la vest și la nord de blocul curent determină starea. OMVQ este similară cu SMVQ, dar blocurile care trebuie codate se potrivesc prin suprapunere, și stările sunt determinate de regiunile de întrepătrundere. Rezultatele[70,69] arată o îmbunătățire a PSNR de 1-3 dB față de schemele de compresie VQ standard, pentru câteva imagini. Oricum, calitatea perceptuală a imaginilor folosind SMVQ și OMVQ arată o îmbunătățire esențială a calității imaginii.

Cuantizarea vectorială a transformatei TVQ (= *Transform Vector Quantization*)[162] aplică VQ coeficienților de transformare ai semnalului original de intrare. Se aplică datelor de imagine o transformare bazată pe blocuri, cum este DCT, pentru a face datele statistic independente sau măcar necorelate și pentru a forța exprimarea energiei semnalului în cât mai puțini coeficienți posibil. Un cadru natural pentru TVQ este aplicarea unei transformări bidimensionale datelor de imagine, clasarea blocului transformat într-una dintre câteva categorii și aplicarea unei matrice de codare VQ bazate pe clasificare.

Codarea trunchiată a blocurilor, BTC (*Block Truncation Coding*),[38,58] se încadrează, de asemenea, în categoria cuantizării vectoriale. Aici, imaginea este divizată în blocuri și fiecare bloc este transferat în două niveluri. Nivelurile sunt alese astfel încât să păstreze primele două momente de eșantionare din bloc. Fiecare bloc este codat cu m și σ, media eșantioanelor, respectiv deviația standard și un vector binar transferă fiecare valoare din bloc într-unul dintre cele două niveluri l_1 și l_2, date de

$$l_1 = m - \sigma\sqrt{\left[\frac{k_2}{k_1}\right]}, \qquad (26)$$

$$l_2 = m - \sigma\sqrt{\left[\frac{k_1}{k_2}\right]}, \qquad (27)$$

unde k_1 este numărul de pixeli din bloc, cu valori mai mici sau egale cu m, iar k_2 este numărul de pixeli din bloc, cu valori mai mari decât m.

O altă abordare a cuantizării vectoriale a fost introdusă inițial pentru a coda datele foarte structurate și rare, găsite în benzile de frecvențe mai înalte ale unui sistem de codare bazat pe subbenzi.[e] Acestă tehnică este cunoscută ca GVQ (*Geometric Vector Quantization* = cuantizare vectorială geometrică)[144,142,139] și poate fi utilizată în general pentru a coda orice date care sunt structurate. Observând figura 4, să notăm că datele de imagine din benzile de frecvență ridicată sunt foarte rare și structurate. GVQ exploatează aceste proprietăți și codează datele de frecvență înaltă foarte eficient.[144,142,139] Această tehnică impune un număr fix de niveluri de cuantizare pentru o anumită dimensiune de bloc. Această tehnică este similară în concepție cu codarea imaginii cu tipare vizuale[21,22f] unde un mic set de tipare geometrice locale sunt folosite pentru a coda imaginea. Amândouă metodele exploatează sensibilitatea ochiului uman la schimbări susținute de intensitate așa cum se întâmplă la un contur.[102] Noua metodă GVQ, atunci când este aplicată cu două niveluri, este, de asemenea, similară cu codarea blocurilor trunchiate la momente absolute de timp, AMBTC (= *Absolute Moment Block Truncation Coding*)[35,58]. Metoda implementării GVQ descrisă aici este bazată pe minimizarea erorii medii pătratice (*mse*), dar o funcție de cost perceptuală ar putea asigura rezultate mai bune.

Notăm faptul că datele din subbenzile ilustrate în figura 4 au anumite structuri asociate cu fiecare poziție a benzilor de frecvență. O matrice de codare pentru forme geometrice simple poate fi suficientă în reprezentarea unor astfel de date. GVQ de L niveluri codează blocuri de date impunând ca fiecare bloc să fie reprezentat prin nu mai mult decât L valori de intensitate. Informația transmisă include indexul k_{best} al vectorului ales din matricea de coduri, reprezentând cea mai bună formă geometrică, precum și nivelurile calculate ale intensităților de ieșire l_1 până la l_L. Această tehnică poate fi gândită ca o formă specifică de cuantizare vectorială cu „câștig în formă". Rata de bit pentru GVQ este redusă prin codarea Huffman a indexului și a nivelurilor de cuantizare și prin implementarea GVQ cu niveluri predeterminate de cuantizare. GVQ restrânsă (*constrained GVQ*)[141] utilizează informații *a priori* despre datele de semnal, cu scopul de a restrânge nivelurile de cuantizare. Datorită faptului că în subbanda de înaltă frecvență datele sunt rare, GVQ este în situația de a avea un nivel la zero.

Cu scopul de a reduce complexitatea căutării celui mai bun vector de cod și a celor L niveluri, este propusă o metodă de căutare rapidă[143] pentru GVQ, care se implementează după cum urmează. Fiecare vector de intrare de lungime N este sortat după valoarea intensității

$$x_1 \le x_2 \le \ldots \le x_N \qquad (28)$$

---

[e] Sistemele de codare bazate pe sub-benzi vor fi introduse în subcapitolul 3.3.

[f] Codarea imaginii cu tipare vizuale va fi descrisă în capitolul 4, la codarea bazată pe modele.

Distorsiunea minimă care se poate atinge este dată de

$$D_{\min} = \min_{1 \le j_1 \le \cdots \le j_l \le N} \{D(1, j_1, l_1) + D(j_1 + 1, j_2, l_2) + \cdots + D(j_{(l-1)} + 1, j_l, l_L)\} \quad (29)$$

unde optime (în sensul minimizării erorii pătratice medii) pentru cele L niveluri de cuantizare sunt valorile medii ale intervalelor de cuantizare optimizate

$$l_m = \frac{1}{j_m - j_{m-1}} \sum_{i=j_{m-1}+1}^{j_m} x_i. \quad (30)$$

Rezultatele codării subbenzilor utilizând GVQ pentru codarea subbenzilor de înaltă frecvență[140] cu o rată totală de 48 kbps[g] sunt arătate în jumătatea de sus a figurii 4.

Ultimii ani ne-au arătat progrese considerabile în înțelegerea cuantizării vectoriale, în mod particular în zona algoritmilor de implementare practică și a tehnicilor de cuantizare vectorială imună în prezența erorilor de transmisie. Datorită acestor realizări metodele de cuantizare vectorială sunt în principiu competitive cu alte tehnici mai cunoscute pentru codarea imaginii. Oricum, de vreme ce metodele bazate pe DCT și MC-DCT sunt standarde unanim acceptate, este de presupus că oportunitățile cuantizării vectoriale sunt cele mai mari în domeniul particular al codării cu rată de bit foarte scăzută, unde performanța tehnologiilor uzuale este încă inadecvată, ca și în aplicații unde caracteristicile canalului sunt o provocare la adresa metodelor curente de codare a surselor.

3.3. *Tehnici multi-rezoluție*

Acest capitol va trata schemele de compresie, care se încadrează în categoria comună a unui cadru multirezoluție. Aceasta include ceea ce în literatura tehnică este cunoscut ca și codare de subbenzi și, în general, ca scheme de compresie bazate pe unde mărginite (*wavelet*). Ideea descompunerii unui semnal de imagine în diferite canale de frecvență, înaintea compresiei, poate fi justificată atât pe baza studiilor statistice cât și a celor psiho-vizuale. Descompunerile în canale multifrecvență au găsit multe aplicații în computerele video și în aplicațiile de procesare de imagini și sunt valoroase în înțelegerea unor procese de nivel mic în sistemul vizual uman (HVS). Avantajul utilizării codării bazate pe subbenzi față de transformările bazate pe blocuri este acela că, prin descompunerea semnalului într-o reprezentare multi-rezoluție, evităm o schemă de codare bazată pe blocuri și astfel artefactele (produsele parazite) asociate ei. Reprezentările multi-rezoluție se pretează în mod natural la încorporarea alocării de bit și a cuantizării fundamentate perceptual. Într-o schemă bazată pe subbenzi se dă o mai mare prioritate benzilor de frecvențe joase și, la rate de bit scăzute, efectul imposibilității de a reprezenta cu acuratețe componentele de înaltă frecvență ale semnalului de imagine se manifestă printr-o pierdere a detaliilor de înaltă frecvență în întreaga imagine. Această denaturare este destul de diferită de erorile de parcelare (*tiling*) din cazul algoritmilor bazați pe DCT.

Codarea subbenzilor a fost inițial propusă pentru codarea vorbirii.[32,33,41] Structura de bază cu două benzi pentru codarea subbenzilor este arătată în figura 2 și constă din

[g] Rata de bit de interes este de 64 kbps din care 48 kbps pentru datele video și 16 kbps pentru vorbire și alte date.

trei părți: (i) analiza, (ii) cuantizarea și codarea și (iii) sinteza. În cel mai simplu caz sistemul cu două benzi arătat în figura 2 operează cu o frecvență normalizată de tăiere la $\pi/2$. $H_0(z)$ și $H_1(z)$ reprezintă notația pentru filtrele de analiză trece jos, respectiv trece sus, în timp ce $G_0(z)$ și $G_1(z)$ reprezintă notațiile pentru filtrele de sinteză. Semnalul de intrare este descompus în două semnale reprezentând componentele filtrate trece jos și cele filtrate trece sus, codate în mod adecvat și reconstruite în etajul de sinteză. Dacă ignorăm pasul cuantizării și codării, putem exprima ecuația de sinteză pentru o jumătate de bandă ca:

$$\begin{aligned}\hat{X}(z) &= \frac{1}{2}[H_0(z)G_0(z)+H_1(z)G_1X(z)]+\frac{1}{2}[H_0(-z)G_0(z)+H_1(-z)G_1X(-z)] \\ &= T(z)X(z)+E(z),\end{aligned} \tag{31}$$

unde $T(z)$ este notarea pentru *funcția de transfer* a sistemului de analiză-sinteză iar $E(z)$ reprezintă *termenul de eroare* (*aliasing*). $T(z)$ trebuie să fie o întârziere pură pentru a nu avea distorsiuni de amplitudine și fază. Dacă ieșirea unui sistem ideal este identică cu intrarea, filtrele se bucură de proprietatea de *reconstrucție perfectă*. Impunând filtrelor de sinteză condițiile de simetrie a imaginii oglindă

$$\begin{aligned}G_0(z) &= H_1(-z), \\ G_1(z) &= -H_0(-z),\end{aligned}$$

filtrele satisfac proprietatea de anulare a erorii (*aliasing*), în absența zgomotului de cuantizare sau de codare. Filtrele sunt, de asemenea, proiectate în așa fel încât să minimizeze distorsiunile de amplitudine și fază, aducându-l pe $T(z)$ cât se poate de aproape de o întârziere pură. O clasă de filtre populare, care au fost proiectate pentru codarea de subbandă, sunt bancurile de filtre oglindă în cuadratură (QMF= *Quadrature Mirror Filterbank*).[33,64,30,175] Filtrul trece sus de analiză H_1 este ales astfel încât să aibă o simetrie a imaginii oglindă în raport cu filtrul trece jos de analiză H_0, rezultând un banc de filtre ortogonal. Bancurile de filtre oglindă în cuadratură (QMF) propuse în [64] sunt utilizate extensiv pentru compresia vorbirii și a imaginii și sunt forțate să aibă fază liniară. Din această constrângere, de a avea simetrie în oglindă și fază liniară, filtrele nu mai pot asigura o reconstrucție perfectă. Oricum, coeficienții lui H_0 pot fi obținuți în așa fel încât caracteristica funcției de transfer să fie făcută cât este posibil de plată, păstrând în același timp energia benzii de oprire a lui H_0 rezonabil de jos.[64]

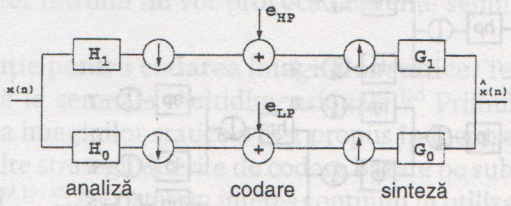


Figura 2. Sistem de analiză / sinteză cu două subbenzi

Proiectarea filtrelor optime pentru reprezentări multi-rezoluție ale semnalelor în vederea diferitelor aplicații a fost studiată pe larg. Activitatea în această arie de interes include bancurile de filtre oglindă în cuadratură (QMF) și proiectarea mai largă a filtrelor bazată pe teoria undelor mărginite (*wavelet*).[33,64,30,170,155,28,175,147,36,37] O prezentare excelentă a proiectării filtrelor de subbandă este făcută în lucrarea [176]. Altă problemă interesantă

de cercetare care poate fi în mod special utilă pentru compresia de imagine şi video este proiectarea filtrelor bidimensionale ne-separabile. În teorie, gradul mai mare de libertate inerent proiectării bidimensionale permite proiectarea unor caracteristici mai bune ale filtrelor, ca şi abilitatea de a proiecta filtre similare cu proprietăţile de selectivitate în frecvenţă ale sistemului vizual uman.[77,67,25,168] Algoritmii de compresie utilizaţi în mod curent continuă să folosească filtre separabile, cu alte cuvinte, filtre unidimensionale aplicate pe fiecare din direcţiile verticală şi orizontală, datorită limitărilor de optimizare a algoritmilor în proiectarea filtrelor bidimensionale.

Deşi s-a lucrat mult la proiectarea filtrelor optime pentru sistemele de codare bazate pe subbenzi, mare parte din această muncă a ignorat efectele pasului codare / cuantizare. A proiecta filtre pentru un sistem perfect de analiză / sinteză poate fi suficient pentru codoare operând la rate de bit înalte, dar dacă filtrele de analiză / sinteză pot fi optimizate în special pentru sistemul de codare şi pentru caracteristicile semnalului la rate de bit mai scăzute, se poate obţine un câştig semnificativ. Baza de lucru descrisă în lucrarea [189] examinează erorile de cuantizare într-un sistem cu subbenzi utilizând QMF, cu cuantizori Lloyd-Max, şi un model cu câştig mare în condiţii de zgomot aditiv (*gain-plus-additive-noise model*). Pornind de la modelul din [189], proiectarea filtrelor de sinteză în prezenţa zgomotului de cuantizare este descrisă în [76]. Experimentele menţionate mai sus constituie încercări preliminare pentru optimizarea filtrelor de analiză şi sinteză în prezenţa efectelor de cuantizare. Proiectarea concomitentă a filtrelor de analiză / sinteză a imaginii şi a algoritmilor de cuantizare continuă să fie o problemă provocatoare, dar nu complet stăpânită încă.

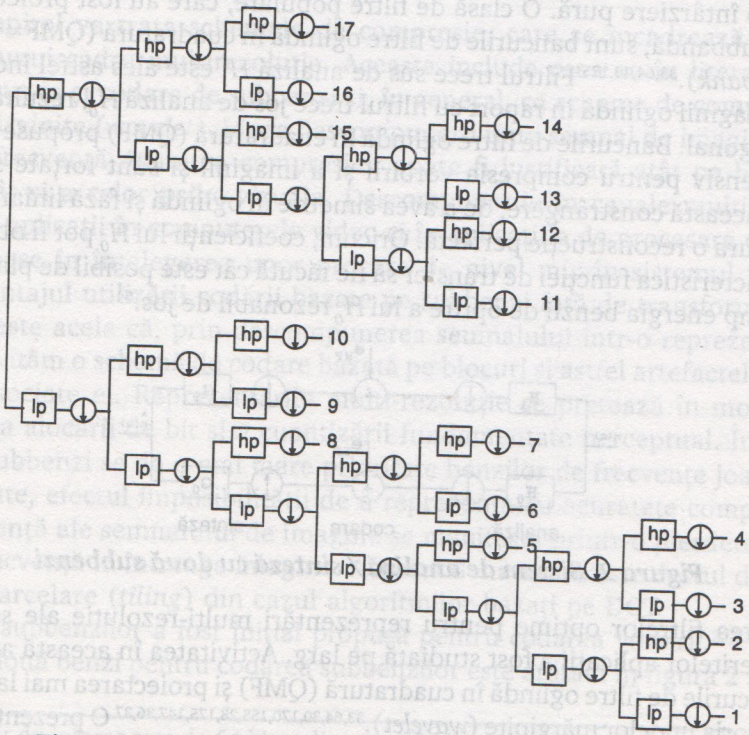


Figura 3. Descompunerea tridimensională în subbenzi

Ideea descompunerii unei secvențe de imagine în benzi de frecvență devine un instrument și mai puternic când considerăm o abordare arborescentă, în care subbenzile pot fi la rândul lor descompuse în benzi inegale. Un exemplu de astfel de descompunere este prezentat în figura 3 pentru o descompunere tridimensională a semnalului video, temporală și spațială. Ideea de bază este de a descompune semnalul original de imagine într-o reprezentare cu rezoluție bună de frecvență la frecvențe joase și rezoluție bună în timp (spațiu) la frecvențe înalte. Flexibilitatea de a schimba rezoluția în timp (spațiu) și pe cea în frecvență este dorită pentru analiza semnalelor nestaționare. Intuitiv, acest tip de descompunere este cel mai util pentru semnale care sunt compuse din componente de joasă frecvență cu durată lungă și componente de înaltă frecvență cu durată scurtă – situație care în general este adevărată pentru multe semnale din practică.[153] Teoria undelor mărginite (*wavelet*) furnizează o teorie unificatoare atât pentru reprezentările multi-rezoluție ale semnalelor cât și pentru proiectarea filtrelor care se folosesc pentru descompuneri.[36,37,97,98] Aici prezentăm unele din diferitele caracteristici de proiectare a filtrelor, împreună cu relevanța lor pentru compresia imaginii.[141] Deși literatura specializată în proiectarea filtrelor insistă adeseori asupra proprietății de reconstrucție perfectă, ea nu este critică pentru compresia imaginii unde distorsiunile de compresie au un efect mult mai puternic. Este suficientă proiectarea filtrelor cu proprietatea de reconstrucție aproape perfectă. Regularitatea este o altă proprietate care nu trebuie să fie accentuată în mod strict. În schemele de codare a imaginii, unde se aplică o descompunere ierarhică pe benzi-octave, sunt suficiente filtre care sunt aproximativ regulate. Caracteristicile dorite ale răspunsului în frecvență depind de schema de compresie și nu se pot aplica reguli generale. Când schema de compresie se bazează pe exploatarea corelațiilor dintre frecvențele benzii, sunt alese filtrele cu răspunsuri mai puțin ascuțite în frecvență. S-a descoperit că faza liniară este una din caracteristicile importante ale filtrelor pentru aplicațiile de compresie de imagine deoarece distorsiunile de fază care apar în jurul contururilor imaginii sunt, din punct de vedere al percepției, deosebit de supărătoare. Pe baza celor de mai sus, filtrele din lucrarea [64], care au o reconstrucție aproape perfectă, sunt aproximativ regulate și au faza liniară, sunt larg utilizate în schemele de compresie bazate pe subbenzi a imaginii. Aceste observații pot fi utilizate ca linii generale de ghidare pentru alegerea filtrelor eficiente pentru codarea semnalelor video și de imagine. În general, atâta timp cât abordarea proiectării filtrelor întrunește condițiile anterioare, variațiile caracteristicilor filtrului nu vor provoca diferențe semnificative în rezultatele compresiei.

Cadre multi-rezoluție pentru codarea imaginilor statice. Teoria descompunerii în subbenzi a fost extinsă la semnale multidimensionale.[180] Primul codor bazat pe subbenzi pentru compresia imaginilor statice a fost propus în lucrarea [193]. De atunci au fost examinate mai multe strategii diferite de codare bazate pe subbenzi, pentru codarea datelor de imagine.[169,189,191,82] A existat un interes continuu în utilizarea compresiei bazate pe subbenzi pentru codarea imaginii, în special simultan cu activitățile recente, legate de reprezentări multi-rezoluție utilizând filtre bazate pe unde mărginite (*wavelet*).[153] În zona compresiei de imagine bazate pe subbenzi sau pe unde mărginite, au fost investigate multe idei interesante. În lucrarea [190] se propune o schemă de codare bazată pe o descompunere pe subbenzi și cuantizare vectorială dar nu în cadrul unei subbenzi, ci peste subbenzi. Această schemă exploatează dependența dintre datele subbenzilor. O schemă de codare bazată pe unde mărginite, prezentată în lucrările [167], [166], introduce

conceptul EZW (= *Embedded Zerotree Wavelet*). Întrucât cea mai mare parte a datelor de imagine prezintă un răspuns de tip trece jos în frecvență, este improbabil ca datele de imagine dintr-o locație spațială particulară să aibă o energie de înaltă frecvență semnificativă dacă componentele corespunzătoare de frecvență mai joasă sunt nesemnificative. Conceptul EZW exploatează acest fenomen codând datele din subbandă într-un canal de frecvență joasă ca rădăcină de arbore-zero (*zero-tree root*) dacă acea dată și toate datele corespondente din aceeași locație spațială dar din benzile de frecvențe mai mari sunt nesemnificative. Pentru cazul unui punct cu date nesemnificative de frecvență joasă și puncte cu date semnificative la frecvențe mai înalte, datele de joasă frecvență sunt codate ca *zerouri izolate*. Această tehnică este foarte atractivă întrucât nu necesită pregătire sau cunoștințe *a priori* despre statistica imaginii. O schemă de compresie bazată pe unde mărginite (*wavelet*) pentru imagini statice[7] utilizează o descompunere în unde mărginite și o cuantizare vectorială (VQ) cu o matrice de coduri (*codebook*) multi-rezoluție pentru compresia datelor.

Cadre multi-rezoluție pentru codarea semnalelor video. S-au realizat multe lucrări interesante utilizând codarea tip subbandă pentru compresia video. O mare parte din această muncă a studiat utilizarea descompunerii bidimensionale a subbenzilor în spațiu, cu compensarea mișcării pentru componentele temporale. Exemplele includ activitatea descrisă în lucrarea [47] pe un codor de 64 kbps, unde semnalul diferență de cadru (cu compensare de mișcare) este descompus utilizând o descompunere bidimensională de subbenzi, structurată arborescent și cuantizată. În lucrarea [191] autorii propun o schemă de codare video bazată pe o descompunere bidimensională în subbenzi a semnalului diferență de cadru. Codorul din lucrarea [71] combină predicția ierarhică cu compensarea mișcării, cu codarea bidimensională cu limitare de entropie.

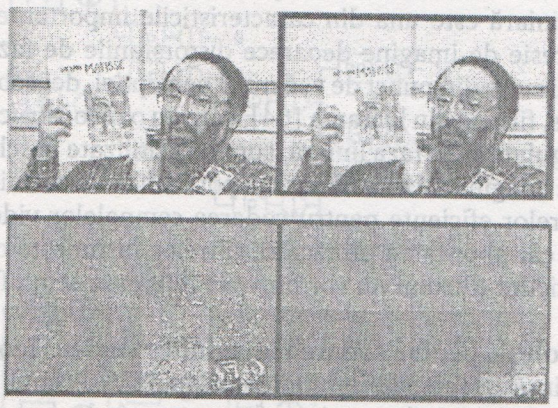


Figura 4. Sus: imaginea originală (stânga); rezultatul codării folosind un codor de subbenzi tridimensional la 48 kbps (dreapta) Jos: decompunerea în 17 subbenzi

O cercetare originală asupra descompunerii și codării unei secvențe video utilizând o descompunere de subbandă tridimensională este prezentată în lucrarea [66]. Pentru codarea imaginii și a semnalului video este uzuală descompunerea spațială a imaginii în subbenzi, utilizând filtre unidimensionale, separate pe direcțiile verticală și orizontală. Această idee poate fi în mod ușor extinsă la trei dimensiuni – orizontală, verticală și temporală – pentru codarea unei secvențe video. Ca un exemplu, lucrarea [140] descompune o secvență originală de imagine în 17 benzi de frecvență după cum este arătat în structura arborescentă din

figura 3. Câteva exemple de secvenţe video descompuse în benzi de frecvenţă conform schemei din figura 3 sunt arătate în jumătatea de jos a figurii 4 unde partea stângă a imaginii reprezintă banda de frecvenţe temporale înalte iar partea dreaptă reprezintă banda de frecvenţe temporale joase. Colţul din dreapta mai jos reprezintă banda cu cea mai joasă frecvenţă spaţială. Alte lucrări bazate pe o descompunere tridimensională includ Referinţele [131],[13],[83], [174], [127], [94]. În lucrarea [143], secvenţa video este descompusă în benzi de frecvenţe spaţio-temporale şi datele de joasă frecvenţă sunt codate folosind cuantizare vectorială structurată arborescent ne-echilibrată (UTSVQ = *Unbalanced Tree Structured VQ*)[96] şi codare inter-bloc fără zgomot (IBNC = *InterBlock Noiseless Coding*)[125] în timp ce datele structurate în benzile de frecvenţă înaltă sunt codate utilizând cuantizarea vectorială geometrică (GVQ = *Geometric Vectorial Quantization*).[h] Schema de codare video tridimensională din lucrările [140], [141] constă din GVQ în benzile de frecvenţă mai înaltă şi o versiune modificată a realimentării condiţionale (*conditional replenishment*) pentru benzile de frecvenţe temporale joase. Un rezultat la 48 kbps utilizând această schemă tridimensională de sub-benzi este ilustrată în partea de sus a figurii 4. Sistemul tridimensional propus în [131] este realizat printr-o descompunere de frecvenţe temporale constând din valorile unui cadru în banda de frecvenţă temporală joasă şi din diferenţa de cadru deplasat în banda de frecvenţe temporale înalte. Pentru a coda datele din subbenzi este aplicată cuantizarea adaptivă cu latice de vectori (*adaptive lattice vector quantization*).

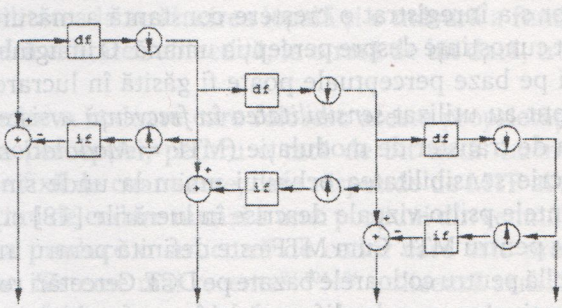


Figura 5. Descompunere piramidală

Codarea pe bază de subbenzi pentru vorbire şi compresia imaginii este bazată pe eşantionarea critică, cu alte cuvinte, numărul de eşantioane după etajul de analiză este egal cu numărul eşantioanelor originale. Proprietatea de eşantionare critică a unui sistem de subbenzi impune constrângeri stricte proiectării filtrelor. O altă abordare pentru reprezentările multi-rezoluţie ale unui semnal este de a permite redundanţă sau supra-eşantionarea semnalului în etajul de analiză, ceea ce permite mult mai puţine cerinţe stricte în proiectarea filtrelor de analiză / sinteză. Aceasta este abordarea folosită în descompunerea piramidală introdusă în lucrările [16], [15]. Structura piramidală de bază este arătată în figura 5. Semnalul original de intrare este descompus într-un semnal x_{LP}, conţinând componentele de joasă frecvenţă ale lui x, şi x_{HP}, conţinând detaliile de înaltă frecvenţă ale lui x. *if* şi *df* corespund filtrelor de interpolare, respectiv decimare. Piramida cu un etaj arătată în figura 5 poate fi iterată într-un model arborescent pentru a obţine o reprezentare cu multiplă rezoluţie a semnalului original. Pentru mai multe detalii asupra teoriei undelor mărginite şi a codării de subbenzi, este utilă trimiterea la lucrările [182], [177], [192].

[h] GVQ este descrisă în capitolul 3.2.

4. Tehnici de compresie bazate pe modele

Acest capitol va introduce unele tehnici de compresie neconvenționale ale căror elemente comune le situează în categoria tehnicilor bazate pe modele. Sunt două distincții importante care trebuie făcute aici – există metode bazate pe modele ale sistemului vizual uman (HVS = *Human Visual System*) și metode bazate pe modele ale imaginilor însele. Metodele bazate pe modele ale HVS conțin uzual unele tehnici convenționale ca aceea de tipul bloc DCT, sau cea a descompunerilor multi-rezoluție (discutată în capitolul 3) cu o schemă de alocare de bit *perceptuală*. Metodele bazate pe modele ale imaginilor includ unele modele foarte restrictive ale imaginilor cum ar fi modelele de caroiaj folosite pentru imaginile video „cu bust", ca și unele modele mai generale, cunoscute sub denumirea de „generația a doua de codare a imaginii".

4.1. *Metode bazate pe modele vizuale*

În general, utilizatorul final al unui sistem de comunicare cu compresia imaginii este o ființă umană. Astfel capătă sens să încercăm a ține cont de caracteristicile receptorului uman în proiectarea sistemelor de compresie a semnalelor de imagine și video. Dacă sistemul vizual uman ar fi utilizat eroarea medie pătratică drept măsură a distorsiunii, sarcina ar fi fost ușoară. Din păcate, sistemul vizual nu este deloc atât de simplu.[48]

De-a lungul anilor s-a înregistrat o creștere constantă a măsurii în care compresia imaginii a încorporat cunoștințe despre percepția umană. O imagine generală a stadiului actual al compresiei pe baze perceptuale poate fi găsită în lucrarea [62]. Majoritatea cercetărilor de început au utilizat *sensibilitatea în frecvență* a sistemului vizual uman, descrisă prin funcția de transfer de modulație (MTF = *Modulation Transfer Function*). Această funcție descrie sensibilitatea ochiului uman la unde sinusoidale de diverse frecvențe. Experimentele psiho-vizuale descrise în lucrările [18] și [101] au propus un model utilizat curent pentru MTF. Cum MTF este definită pentru intrări sinusoidale, ea nu este direct utilizabilă pentru codoarele bazate pe DCT. Cercetări recente[128] au dezvoltat o transformare care ține seama de diferențele între fundamentele Fourier și DCT. Proceduri similare pot fi utilizate pentru transformarea modelului în alte cadre pentru analiza în frecvență. Din aceste modele, dat fiind faptul că distanța minimă de vizualizare este fixă, este posibilă determinarea unui prag static de *diferență abia sesizabilă* (JND = *Just Noticeable Difference*) pentru fiecare bandă de frecvență. Aceste praguri pot fi utilizate atât pentru cuantizare[183] cât și pentru alocarea de bit[137]. Conceptul determinării pragului JND pentru fiecare locație dintr-o imagine este un concept central pentru multe dintre abordările perceptuale pentru compresia imaginii.

Modelele pot fi extinse pentru a include și *sensibilitatea la strălucire*. Sensibilitatea la contrast este o cale de a măsura efectul detectării pragului de zgomot pe un fundal constant. Acest fenomen depinde de nivelul de gri al fundalului ca și de nivelul de contrast al zgomotului. Pentru sistemul vizual uman această funcție este una neliniară. Sensibilitatea la contrast poate fi inclusă atât implicit, prin dezvoltarea unor funcții ale raportului între pragul de mascare și strălucirea locală, sau explicit, utilizând sisteme omomorfice sau așa numitele *spații perceptual uniforme* sau *spații color echi-luminante*.[179]

Deși metodele mai sus menționate reprezintă un punct de pornire bun în utilizarea proprietăților sistemului vizual uman pentru compresia imaginii, ele nu folosesc în întregime proprietățile de mascare ale sistemului vizual uman. Sensibilitatea în frecvență

este o proprietate globală depinzând doar de mărimea imaginii și condițiile de vizualizare. Sensibilitatea la strălucire a fost exploatată în principal prin pre- și post-procesarea întregii imagini. În mod ideal, avem nevoie de un model mai dinamic care să permită un control mai fin al procesului de cuantizare.

În plus față de aceste efecte, o altă proprietate folositoare pentru compresie este *mascarea vizuală*, cunoscută de asemenea în literatura de specialitate sub denumirea de mascare a contrastului.[172] Mascarea este un fenomen în care detectibilitatea componentei unui semnal depinde de prezența sau absența altor componente ale semnalului în vecinătatea imediată. Acest proces poate fi înțeles prin următorul exemplu. Considerăm o imagine compusă dintr-un nivel constant de gri. Schimbând nivelul de gri al unui singur pixel la o valoare peste pragul de detecție a contrastului, acest pixel va fi destul de vizibil. Înlocuind imaginea de nivel constant de gri cu una care constă dintr-un zgomot aleator cu același nivel mediu de strălucire ca în imaginea plată, devine extrem de dificil să localizăm acel pixel singular situat deasupra pragului de detecție a contrastului. În acest caz, prezența zgomotului ca semnal de fond *maschează* prezența celui de-al doilea semnal.

Toate aceste proprietăți depind de conținutul scenei locale și se poate astfel spune că au proprietatea, dorită, de *control local*. Cu alte cuvinte, nivelul local de *diferență abia sesizabilă* (JND) depinde de funcția de transfer de modulație (MTF) a sistemului vizual uman, sensibilitatea la strălucirea locală și o măsură a fundalului local. Pentru a atinge cea mai înaltă calitate subiectivă, la o rată de bit dată, trebuie utilizate toate aceste proprietăți.

Au fost propuse câteva codoare care folosesc aceste proprietăți, pentru a atinge o calitate mai înaltă a percepției, pentru o rată de bit dată. Lucrarea [26] utilizează conceptul de matrice fixă de cuantizare, care răspunde de MTF al sistemului vizual, în conjuncție cu conceptul de codare numai a unei porțiuni din semnal, cea care depășește pragul local. Aceasta produce o structură de codor care este precursorul codoarelor perceptuale moderne. Sisteme mai recente[27,104,126] utilizează sensibilitatea la frecvență și strălucire și, în plus, folosesc anumite forme de mascare ale texturii locale. În aceste metode, în loc de determinarea explicită a pragurilor JND – încorporând frecvența, contrastul și textura, fiecare bloc DCT este clasificat în funcție de energia sa. Pe baza acestei clasificări se apelează la cea mai potrivită regulă de cuantizare.

Alegerea bancului de filtre de analiză utilizat în codor poate afecta performanța sistemului de compresie. Scopul principal al cuantizării perceptuale este de a controla poziția spațială și în frecvență a distorsiunii de cuantizare. În această idee, bancul de filtre trebuie să fie ales astfel încât să permită această formă de control. La modul ideal, însumarea distorsiunii de cuantizare la un coeficient trebuie să nu se manifeste în coeficienți care nu sunt adiacenți celui care a fost perturbat. Bancul de filtre de analiză trebuie să imite structura sistemului vizual. Pentru o ființă umană, această structură este un set de filtre cu spațierea frecvențelor de aproape 1.5 octave și o deschidere unghiulară de aproximativ 40 grade.

DCT și alte bancuri de filtre uniforme satisfac primul criteriu,[72] dar nu și pe al doilea. Aceasta conduce la dificultăți în crearea modelelor de mascare, deoarece este o nepotrivire între structura de bază a modelului și structura transformării care este implementată. Un mijloc de a implementa o transformare care imită sistemul vizual este *Transformarea Cortex*[187] și aceasta se concretizează într-un codor de imagine bazat

pe această transformare[186]. Principalul obstacol în această aplicație este faptul că Transformarea Cortex nu este un banc de filtre cu decimare maximală. Aceasta se materializează într-o extindere a cantității de date care trebuie să fie codată. În prezent, modelele de mascare vizuală nu sunt capabile să furnizeze suficient câștig de codare pentru a depăși acest dezavantaj.

După cum s-a mai punctat anterior, raportul semnal-zgomot nu este un predictor de acurateţe a calității subiective a imaginii, în special la rate de bit joase.[1,14,42,86,119,154] O cale de rezolvare a acestei probleme este transformarea erorilor într-un spațiu unde poate fi folosită eroarea medie pătratică. Aceasta este abordarea prezentată în lucrările [128] și [161]. Imaginile originale și codate sunt transformate într-un „spațiu vizual uman", utilizând MTF și Legea lui Weber. Această transformare este proiectată să compenseze efectul acelor porțiuni ale sistemului vizual. Eroarea medie pătratică dintre imaginile transformate furnizează o distorsiune ponderată. O astfel de abordare produce o corelație mai bună a evaluării subiective decât eroarea medie pătratică neponderată, dar este limitată de faptul că modelul vizual este global în natură. Pentru a mai ușura această problemă, lucrarea [35] introduce conceptul unui predictor vizual de diferență. Un model vizual detaliat este aplicat imaginii originale și pragurile locale JND sunt calculate pentru fiecare pixel din imagine. Diferențele dintre imaginile originale și cele de după compresie sunt apoi calculate și transformate în unități JND. Reprezentarea rezultată dă o reprezentare bună a calității imaginii. Porțiunile de la intrare, care au erori de magnitudinea unui JND prezintă degradări mici sau invizibile, în timp ce zonele cu amplitudini mai mari de eroare se corelează cu porțiunile în care imperfecțiunile de codare sunt vizibile. Figura 6 prezintă două exemple de imagine codată, împreună cu profilele corespunzătoare cu erori perceptuale relevante. Algoritmul de codare este JPEG, sistem descris în capitolul 5. Pixelii negri denotă porțiuni ale imaginii unde distorsiunea este la nivelul de 1 JND sau mai puțin. Pixelii de alb pur denotă erori la nivelul de 15 unități JND. Pentru comparație, PSNR la imaginea din stânga este de 34 dB, în timp ce la imaginea din dreapta este 31 dB.

Un codor mai eficient poate fi generat prin utilizarea unui model perceptual local bun pentru a identifica porțiunile nerelevante ale unui semnal, lucrând în combinație cu un sistem eficient de înlăturare a redundanței. Acum va fi prezentată exact această abordare a structurării codorului în jurul generării unui profil de diferență minimă observabilă (MND = *Minimal Noticeable Difference*), urmată apoi de cuantizarea informației rezultate.[75,158,159]

Codorul de imagine bazat pe JND. Punctul de pornire pentru codorul perceptual adaptiv local este descompunerea bidimensională uniformă în subbenzi, care împarte imaginea originală în 16 benzi uniforme de frecvență. Blocurile de imagine din fiecare subbandă sunt apoi analizate pentru determinarea valorilor locale ale strălucirii și ale texturii.

Profilele de sensibilitate la distorsiuni sunt apoi derivate ca funcții de frecvență, strălucire și textură[158] și parametrii de culoare.[159] Aceasta se realizează utilizând paradigma (modelul) experimentală de bază în care un subiect vizualizează un careu de zgomot cu putere variabilă la o distanță specificată și determină variația zgomotului pentru care careul de zgomot devine *abia observabil* față de un fundal specific, sau o textură plată (uniformă) de imagine. Mai exact, zgomotul aleator este adunat la banda de frecvență în chestiune și imaginile afectate sunt trecute prin bancul de filtre de sinteză. În experimentele descrise, zgomotul are aproximativ aceeași valoare medie ca

Figura 6. Sunt prezentate aici două imagini codate cu standardul de compresie de imagine JPEG, şi o reprezentare a erorilor perceptuale pentru fiecare dintre ele. Cu cât pixelii sunt mai albi în imaginea de eroare, cu atât eroarea este mai perceptibilă în acea locaţie.

şi semnalul de fundal şi funcţia sa de densitate era uniformă. Experimentul este repetat la diferite niveluri de strălucire ale fondului pentru a se determina *sensibilitatea la strălucire*. Când valoarea medie a careului de zgomot este aceeaşi cu cea a fondului, careul de zgomot tinde să devină mai vizibil faţă de fondul gri-mediu, în ipoteza că imaginea de intrare include corecţia gamma.[120] Sensibilitatea la frecvenţă este măsurată experimental prin repetarea testului subiectiv de bază pentru fiecare din cele 16 subbenzi. În final, *sensibilitatea la textură* este determinată printr-un calcul empiric în care textura este definită ca o varianţă intra-bloc cu o frecvenţă locală dominantă şi se foloseşte (din nou) MFT pentru a stabili o relaţie între vizibilitatea distorsiunii şi textură.

Interacţiunea între diverşii parametri care controlează sensibilitatea la zgomot este mică. De aceea, se pot suprapune efectele componente pentru a prezice vizibilitatea totală a zgomotului ca o funcţie a diferiţilor parametrii ai semnalului de intrare.

Datele din experimentul generic de mai sus sunt folosite pentru a deduce pragurile JND, estimate ca profile spaţiale ale fiecăreia dintre cele 16 subbenzi ale imaginii ce trebuie codate. Aceste profile JND conduc la rândul lor la alocări de bit adaptive spaţial pentru fiecare subbandă. Pentru o imagine tipică, unora dintre blocurile spaţiale li se alocă zero biţi şi de fapt, un număr mare de subbenzi primesc o alocare de zero biţi pentru *toate* blocurile lor spaţiale.

Acest tip de abordare poate fi folosit pentru a îmbunătăţi codecurile JPEG.[156]

Codarea perceptuală a facsimilului. Tehnologiile convenţionale de codare de facsimil folosesc o împărţire în două nivele a tonurilor imaginii la transmiţător şi folosirea unor tehnici cum ar fi codarea cu lungime variabilă pentru a comprima datele binare rezultate

din imagine. Un exemplu al acestui tip de tehnici este algoritmul standard JBIG descris în lucrarea [108]. În continuare vom descrie avantajul transmiterii imaginilor de facsimil prin folosirea codării perceptuale a nivelelor de gri originale.

Ratele joase de bit oferite de codarea perceptuală a imaginii sugerează o nouă abordare posibilă pentru transmisia facsimilelor cu niveluri de gri. Mai degrabă decât transmiterea unei fotografii cu niveluri de gri prin împărțirea în prealabil a tonurilor în două și transmiterea fotografiei ca document alb-negu cu o mare rezoluție spațială, se pot transmite *imagini cu trepte de gri* folosind codarea de subbenzi și realizând împărțirea în două tonuri *la receptor*. Au fost comparate două alternative specifice: transmisia fax convențională a imaginii bi-tonale de 1536×1536, folosind tehnicile de codare standard alb-negru; și noua metodă de transmisie a unei imagini 512×512 cu niveluri de gri și folosind împărțirea bi-tonală la recepție. Algoritmul de compresie a fost ales în ambele cazuri fără pierderi, cu un criteriu perceptual de absență a pierderilor pentru metoda subbenzilor. Metoda de transmisie a nivelurilor de gri a condus la o scădere cu 50% a ratei de bit. Mai semnificativ, disponibilitatea informațiilor de gri în noua tehnică permite folosirea unei noi tehnici bazate pe modele pentru împărțirea bi-tonală eficientă la receptor. Modelele folosite includ modelul HVS și modelul imprimantei.[124,135,134]

Compensarea mișcării urmată de 2-D DCT. Compresia video este un domeniu de aplicații unde folosirea indicațiilor perceptuale este puțin înțeleasă. Abia am început să folosim metodologia JND pentru proiecatera cuantizatorilor. Pe un nivel diferit, chiar înainte de cuantizare, folosim încă criterii nestandardizate pentru proiecte fundamentale, cum ar fi alegerea rezoluțiilor spațiale și temporale și a gradului de sub-eșantionare a crominanței.

O abordare hibridă a codării video, abordarea MC-DCT constă dintr-un proces diferențial inter-cadre, urmat de codarea în domeniul frecvență a erorilor necompensate inter-cadre. Această metodă stă la baza câtorva standarde internaționale (CCITT și ISO-MPEG) pentru videoconferințe și video adresabil.[81,88] Standardul MPEG Faza 1 folosește o rată de bit de cca. 1 Mbps; codorul CCITT operează la p*64 kbps, cu p variind de la 1 la 24.

Sistemul ISO-MPEG orientat pe stocare reprezintă un subset periodic de cadre într-un mod de *codare intra-cadre* de înaltă rezoluție (sau mod *intra*), în interesul adresabilității. În modul inter-cadre, codorul folosește o compensare a mișcării pentru a genera, pornind de la cadrele precedente, o predicție a cadrului video curent, iar în cazul ISO-MPEG pentru cele viitoare. Este apoi codată diferența între această predicție și cadrul curent. Cadrele diferență sunt codate în codoarele CCITT și ISO-MPEG folosind tehnici similare celor din codorul JPEG descris înainte.

Se folosește în general o matrice globală de cuantizare pentru întreaga scenă care trebuie codată, dar există un anumit grad de libertate disponibil. La nivelul unui macro-bloc (4 blocuri 8×8 de luminanță adiacente) este posibil să se regleze scalarea matricii de cuantizare. Acest parametru de reglaj este introdus pentru a permite codecului să fie de rată constantă, dar poate fi, de asemenea, folosit ca reglaj pentru variațiile locale ale nivelurilor de mascare.

O abordare care încorporează cuantizarea pe baze perceptuale în codoarele DCT cu compensare a mișcării este descrisă în lucrarea [148]. Abordarea constă în modificarea parametrului de scalare a cuantizatorlului pe baza criteriilor perceptuale. Întâi, sunt deduse opt clase de complexitate diferite ale scenelor de intrare și se calculează alocările de bit dorite pentru fiecare dintre aceste clase. Apoi, fiecare macro-bloc este clasificat

ca *textură* sau *detaliu fin*, pe baza variantei proprii. În continuare, blocurile de textură sunt discriminate după cum sunt ele: structurate (de exemplu marginile) sau nestructurate. Pe baza acestei clasificări și a complexității generale a scenei se setează parametrul de scalare a cuantizatorului. Aceasta diferă de abordarea strictă JND/MND prin faptul că pragurile perceptuale nu sunt specific determinate dar sunt implicite prin regulile de reglare.

În sfârșit, se poate, de asemenea, adapta la sistemele CCITT și MPEG o tehnică de pre-cuantizare. Datele care trebuie codate sunt diferențe locale, în timp ce modelul perceptual oferă praguri MND. Dacă diferența locală este mai mică decât pragul MND la acea locație, coeficientul nu mai trebuie transmis. Pentru codoarele de tip MPEG, care funcționează la un nivel de calitate constant, se obține o reducere a ratei de bit între 5 și 35% fără nici o pierdere vizibilă în calitatea imaginii.

Un codor HDTV cu rată de bit redusă, bazat pe codarea perceptuală a erorilor inter-cadre a fost recent descris. Metodologia folosită aici a fost similară cu cea a *codorului de imagine bazat pe JND*, dar extensia a fost aproximată prin aceea că structura erorilor inter-cadre a fost foarte diferită de modelul zgomotului uniform, folosit în alte experimente JND. Mai mult, distanța de privire la HDTV a fost mai mică decât de șase ori înălțimea ecranului, ca în celălalt experiment JND. Aceste efecte sunt compensate datorită faptului că de data aceasta JND poate fi o funcție și de timp, iar beneficiile mascării temporale sunt acum disponibile. În contextul unui nou fundal descoperit există o mică perioadă de întârziere în timpul căreia zonele nou expuse nu trebuie să fie perfect reconstruite.

Imaginile HDTV (1280 × 720 pixeli, 60 de cadre pe secundă) sunt comprimate la 17 Mbps folosind procedura de mai sus, pentru a facilita simularea transmisiei pe canalele NTSC de 6 MHz.[121] Rata de 17 Mbps nu este adecvată pentru a oferi codare perceptuală transparentă pentru toate intrările HDTV. Ca rezultat, sistemul oscilează între modurile JND și MND, cu impunerea unei rate de bit constante în ambele cazuri, fără nici o reclamație asupra optimizării de ansamblu.

4.2. *Metode bazate pe modele de imagine*

Acest capitol va descrie câteva tehnici de compresie de imagine care nu intră în categoria tradițională a codării formelor de undă. Aceste tehnici încearcă să exploateze unele cunoștințe a priori despre structura imaginii, astfel încât să minimizeze volumul de date necesare descrierii imaginii. Acest capitol se ocupă de modele de intensități ale imaginilor, care se bazează pe clasificarea datelor de imagine pe contururi, arii cu semnal constant sau texturi, precum și modele mai structurate de figuri într-o scenă cu „prim plan" (bust). Între ideile care stau la baza tehnicilor pe care le vom discuta în acest capitol există multe întrepătrunderi și, de fapt, unele dintre ideile din acest capitol sunt compatibile cu metodele de codare descrise în alte capitole.

Modele de intensitate. Germenele a ceea ce se numește a doua generație de metode de codare a imaginii este descris în lucrarea [78]. Aceste tehnici sunt puternic cuplate cu proprietăți ale sistemului vizual uman și ar fi putut fi descrise în cadrul categoriei de codoare perceptuale. Abordarea codării piramidale[16] descrisă în capitolul 3.3 poate fi considerată o metodă de codare de imagine de a doua generație datorită descompunerii ierarhice și a funcțiilor care se aseamănă bine cu proprietățile HVS. Metodele care intră în această categorie includ tehnici de segmentare care încearcă să caracterizeze texturile și contururile care definesc diferite regiuni de texturi. Segmentarea care definește contururile se poate face în mai multe feluri. Sunt două tehnici generale pentru a găsi

contururile: segmentarea regiunilor-crescătoare, unde contururile sunt regiuni închise bine definite, și tehnicile de detectare a fronturilor, unde contururile nu sunt închise. Tehnica de detectare a fronturilor induce o sarcină mai dificilă, aceea de a defini texturile odată cu regiunile. Tehnicile regiunilor-crescătoare se bazează pe clasificarea pixelilor ca pixeli de contur sau pixeli de textură, cu condiția ca variația între pixeli într-o anumită regiune să nu conțină discontinuități abrupte.[165,74] Abordarea regiunilor crescătoare constă în izolarea regiunilor cu texturi regulate (semnal cât mai constant) și de a reduce posibilitatea de a obține foarte multe segmente mici. Scopul este de a uniformiza regiunile cu texturi fără a pierde informațiile de contur. Un filtru de gradient invers poate să-și adapteze coeficienții la caracteristicile locale ale imaginii, astfel încât să acționeze ca un filtru trece jos pentru zonele fără contururi și ca un filtru trece tot pentru zonele cu contururi. A doua parte a acestei sarcini este pasul creșterii regiunii care începe *cu un pixel și cu calcularea* unor proprietăți ale sale și apoi determină dacă pixelii din vecinătate se bucură de aceleași proprietăți. Dacă un set de pixeli învecinați au aceleași proprietăți, ei sunt considerați ca aparținând aceleiași regiuni. Această procedură continuă până când toți pixelii sunt asignați unei regiuni particulare. Proprietățile folosite pentru o astfel de abordare pot fi foarte simple, cum ar fi diferențele de nivel ale griului care conduc de obicei la prea multe contururi, sau proprietăți mai complexe care conduc la o mai bună reprezentare a contururilor, dar cer o complexitate mai mare a algoritmului de calcul. Pentru a reduce numărul de regiuni obținute printr-o proprietate simplă ca diferența între pixeli, abordarea include unirea regiunilor mici și unirea regiunilor cu contururi slabe. Procedura regiunilor crescătoare permite introducerea unor informații *a priori* cum ar fi limitarea numărului maxim de regiuni. Fiecare regiune poate fi reprezentată prin funcții analitice cum ar fi curbe sau polinoame.

Contururile date de segmentare trebuie să fie codate eficient. Metoda descrisă în lucrarea [78] codează informațile de contur folosind aproximații de linii, aproximări de segmente de cerc și puncte fixe de îmbinare. Cealaltă parte care trebuie codată este descrierea texturii (fundalului) pentru fiecare regiune. Codarea texturii se poate face în doi pași, așa cum este descris în [78]. Regiunea este descrisă de un polinom bidimensional care dă o aproximare a formei generale din fiecare regiune și dintr-o cantitate adaptabilă de zgomot pseudo-aleator care se adaugă la textura fină și la „naturalețea" care se pierd în etapa de preprocesare. Această metodă a condus la un raport de compresie de 50:1 pentru imagini statice.

O altă abordare a codării de generația a doua se bazează pe detectarea și păstrarea informației de contur și este cunoscută sub denumirea de codare bazată pe descompunerea direcțională.[78] O astfel de abordare are puternice legături cu sistemul vizual uman despre care s-a arătat că scoate în evidență procesarea de păstrarea informațiilor de contur. Imaginea este analizată folosind filtre direcționale, compuse din filtre trece sus pe direcția principală și filtre trece jos de-a lungul direcției ortogonale. Această abordare de a divide imaginea în diferite subbenzi de frecvență este similară cu codarea subbenzilor descrisă în capitolul 3.3. Informația de joasă frecvență este codată adecvat, folosind codarea transformatei. Informația de frecvență înaltă este comprimată prin detectarea și codarea datelor de contur. Datele de contur sunt codate folosind codurile de lungime variabilă și Huffman. Metodele bazate pe codarea contururilor și a texturilor nu conduc la distorsiunile tipice, găsite în urma tehnicilor codării clasice a formelor de undă, cum ar fi „mozaicul" sau „zgomotul de purici" dar poate produce imagini sintetice sau nenaturale.

O abordare interesantă care poate fi descrisă ca bazată pe model, a fost introdusă în lucrările [21], [22]. Metoda de codare, care a primit denumirea de VPIC (=*Visual Pattern Image Coding*), are unele similitudini cu cuantizarea vectorială (VQ),[87] codarea trunchiată de blocuri (BTC = *Block Truncation Coding*)[38] și cuantizarea vectorială geometrică (GVQ)[144] toate descrise în capitolul 3.2. VPIC conduce la un raport de compresie mare, similar cu VQ la un cost de calcul foarte mic. Ideea de bază este de a reprezenta o imagine cu unele tipare geometrice de bază care nu depind de imagine și, prin urmare, nu necesită etapa de formare. Tiparele sunt alese pe baza unor procesări de nivel redus ale sistemului vizual uman și sunt caracterizate ca tipare uniforme sau de contur. VPIC a fost extinsă la versiunea ierarhizată (HVPIC= *Hierarchical VPIC*)[22], unde se aplică VPIC la diferite niveluri de rezoluție, construite folosind descompunerea piramidală Laplace.[16] Codarea imaginii oferită de HVPIC asigură o calitate perceptuală mai bună decât VPIC.

O altă abordare, care nu intră în categoria tehnicilor tradiționale de codare a formelor de undă, este ideea codării bazată pe proprietatea fractală.[60] Ideea este de a exploata asemănările care apar în cadrul aceleiași imagini prin folosirea unei transformări afine, care să transfere un bloc „fiu" într-un bloc „tată". Blocul „fiu" este cel care va fi comprimat în timp ce blocul „tată" este blocul de referință. Imaginea comprimată constă din transformări „tată-fiu" și imaginea decodată este obținută prin iterarea de transformări plecând de la o imagine inițială arbitrară. Calitatea obținută folosind o astfel de tehnică este similară cu tehnica VQ la 0,5 bpp pentru compresia imaginilor statice. De fapt, codarea bloc fractală ca implementare curentă poate fi privită ca o formă de VQ cu mare adaptivitate.

Modele obiect. O altă tehnică de codare bazată pe modele folosește o informație *a priori* despre structura generală a feței unei persoane, pentru a comprima datele video. Acest tip de tehnică poate fi utilă în special pentru video-telefonie la rate de bit foarte scăzute, unde compresia cerută este foarte mare, iar imaginea este în general limitată la scene de bust (cap-și-umeri). Două abordări de bază intră în această categorie. Una utilizează un model tridimensional al feței unei persoane[4] iar celălalt folosește un model bidimensional cu segmentare triunghiulară deformabilă și transformări afine pentru estimarea mișcării.[117]

Pentru video, abordarea bazată pe modele conține de obicei un model de segmentare care poate fi foarte specific, ca un model de suprafață tri-dimensională sau, mai general, ca regiuni sau contururi sau margini flexibile bidimensionale. Tehnicile de codare din generația a doua intră în categoria modelelor bidimensionale. Algoritmii bazați pe modele conțin uzual și un model de mișcare, care poate fi specificat în termeni de mișcare locală sau globală tridimensională, sau prin translații sau transformări afine.

Tehnicile bazate pe modele tridimensionale se împart în două categorii. Recent a apărut un excelent articol – recenzie despre acest subiect.[5] O tehnică modelează suprafața obiectelor folosind forme geometrice de bază. Cealaltă categorie este un model parametric care este urmarea unor foarte interesante rezultate de codare a feței umane. Slăbiciunea sistemului bazat pe modele tridimensionale constă în dezvoltarea unui sistem complet automat, precum și din absența unui model general.

Vom trece pe scurt în revistă etapele de bază ale unui sistem de codare bazat pe modele, care utilizează un model parametrizat. O manieră tipică de a modela fața unei persoane este de a adapta un caroiaj-model generic al unei fețe la „obiectul" care urmează a fi codat și de a transfera textura imaginii originale în modelul-caroiaj modificat. În

vederea îmbunătățirii acestui model au fost propuse câteva tehnici, inclusiv adăugarea unor informații despre profilul obiectului la caroiajul-model. Transferarea texturii în caroiajul-model și sinteza expresiilor faciale sunt o parte importantă a metodei bazate pe model, care sunt necesare pentru a crea fețe care să arate „natural". Analiza expresiilor faciale poate fi realizată folosind o tehnică „clip-and-paste (taie și lipește)"[5], unde regiunile expresive sunt extrase din imaginea de intrare și transmise la decodor, și o „metodă de deformare facială structurată"[5], unde expresia facială este sintetizată prin deformarea 3-D a caroiajului-model. Mișcările faciale și expresiile pot fi parametrizate în câteva feluri, incluzând: *reproducerea nivelului de textură* ca și în metoda „clip-and-paste", *parametrizarea nodurilor* unde nodurile caroiajului-model sunt ajustate, *parametrizarea formelor*,unde se ajustează formele caroiajului-model, și parametrizarea mușchilor, unde modelele mușchilor sunt transferate în caroiajul-model. În sistemele de codare a imaginii metodele cel mai des utilizate pentru parametrizarea facială includ abordarea formei și a texturii. Tehnicile utilizate pentru analiza mișcărilor feței includ urmărirea globală a mișcării capului și urmărirea mișcărilor locale pentru estimarea expresiei faciale.

În categoria tehnicilor bazate pe modele, au fost propuse multe tehnici interesante de codare. Interesul în rate de bit foarte joase pentru aplicații ca video-telefonia precum și activitatea curentă în vederea anticipării sistemului MPEG-IV au condus la o creștere a activității de cercetare a tehnicilor bazate pe modele. Tehnici care exploatează caracteristicile tridimensionale sunt incluse în lucrările [113] și [39], care folosesc o suprafață segmentată, modelată de pete planare sau parabolice. Alte cercetări în acest domeniu[133,109,73] propun o metodă care estimează un model de suprafață lină (cu semnal constant) determinată de secvența imaginilor. Aceste metode au fost folosite împreună cu compensarea mișcării pentru a îmbunătăți performanțele tehnicilor tradiționale de codare a formelor de undă. În categoria codoarelor bazate pe modele tridimensionale folosind modele parametrizate, unele cercetări interesante includ animația imaginilor asistată de vorbire[24] unde imaginea este sintetizată pe baza recunoașterii vorbirii din semnalul audio; tehnici hibride utilizând codarea bazată pe model și codarea formelor de undă ale semnalului de eroare bazat pe model au fost, de asemenea, recent investigate. Aceste metode hibride includ codarea bazată pe modele cu MC/DCT,[115,116] codarea bazată pe modele cu cuantizare vectorială[44] și codarea bazată pe modele cu codarea contururilor.[105] O abordare bazată pe modele tridimensionale, folosind un model parametrizat al feței umane, a fost, de asemenea, propus.[4] În acest sistem, codorul separă fața de fundal, estimează parametrii de mișcare ai feței și expresiile faciale și transmite parametrii analizați. Codorul actualizează și corectează parametrii pe măsură ce imaginea se schimbă în timp. Decodorul folosește parametrii transmiși și un model 3-D al feței pentru a reconstrui cadrele imaginii.

O abordare mult mai generală, care nu depinde de un model atât de rigid, este metoda propusă în lucrarea [117], bazată pe regiuni triunghiulare deformabile și pe transformări afine pentru estimarea mișcărilor. Această tehnică nu este limitată numai la imagini de bust. Implicită în această tehnică este abilitatea de a opera zoom (lupă) și rotații, estimarea mișcării obiectelor tridimensionale și evitarea artefactelor de mozaic asociate cu tehnicile de codare a formelor de undă cum sunt metodele bazate pe DCT/MC. O tehnică hibridă folosind modelul tridimensional împreună cu compensarea mișcării bazată pe triunghiuri bidimensionale a fost investigată în lucrarea [5]. Rezultate foarte promițătoare sunt arătate pentru scenele de bust cu activitate redusă (secvența Miss America) la 5 kbps unde 1 kbps este alocat pentru modelul tridimensional și 4 kbps sunt alocați pentru

estimarea mișcării bazate pe triunghiuri. Extinderea acestor metode la urmărirea obiectelor mai complexe, cum ar fi corpurile, este o problemă incitantă de cercetare.

5. Standarde

Standardele de compresie descrise în acest capitol au fost stabilite de *Joint Picture Experts Group* (JPEG) și de *Motion Picture Experts Group* (MPEG). Standardul JBIG pentru imagini binare aeste descris în lucrarea [108].

5.1. *Standardul de compresie a imaginilor statice JPEG*

Acest standard[136,183] este format în realitate din două sisteme de codare separate. Primul oferă compresia fără pierderi cu un raport de compresie relativ mic, în timp ce al doilea constă dintr-o familie de codecuri cu pierderi, bazate pe tehnica DCT.

Codecul JPEG fără pierderi. Acest codec a fost proiectat pentru a oferi un sistem de codare simplu, dar eficient și fără pierderi. El constă dintr-un set simplu de predictori spațiali urmat de un codor entropic (figura 7). Predictorul se bazează pe regiunea de suport ilustrată în figura 8. Predicția eșantionului curent, X, este dedusă dintr-o combinație a celor trei eșantioane învecinate (A, B și C).

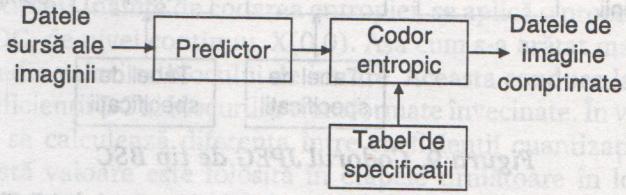


Figura 7. Codorul JPEG fără pierderi

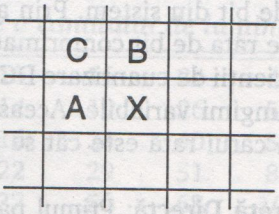


Figura 8. Vecinătatea de predicție pentru codorul JPEG fără pierderi

Tabelul 1. Moduri de predicție pentru JPEG fără pierderi

Valoarea selecției	Predicția
0	Nici o predicție
1	A
2	B
3	C
4	A+B-C
5	A+((B-C)/2)
6	B+((A-C)/2)
7	(A+B)/2

Setul de predictori disponibili este dat în tabelul 1. Această predicție este scăzută din valoarea reală a eșantionului X. Valoarea selecției și această diferență sunt codate fără pierderi folosind fie codarea aritmetică fie codarea Huffman.

În modul de operare fără pierderi, sunt specificate două codecuri, câte unul pentru fiecare opțiune de codare a entropiei. Precizia datelor de la intrare poate varia de la 2 la 16 biți / eșantion. Această abordare produce o compresie de aproximativ 2:1 pentru imagini color cu scene de complexitate moderată.

Codecul JPEG cu pierderi: Toate codecurile cu pierderi se bazează pe transformata cosinus discretă (DCT). Cel mai simplu dintre acestea se numește BSC (= *Baseline Sequential Codec*). Așa cum se arată în figura 9, codorul constă din trei etaje majore, o transformare directă, cuantizatorul și codorul entropic. Transformata directă este DCT 8×8. Scopul ei este de a reduce numărul de eșantioane care trebuie transmise prin realizarea unei compactizări a energiei semnalului. Deoarece cea mai mare parte a imaginilor are un spectru de frecvențe joase, transformarea domeniului spațial în domeniul frecvențelor conduce la un număr mai mic de eșantioane semnificative. În plus, aceste eșantioane tind să se înghesuie în zona frecvențelor joase.

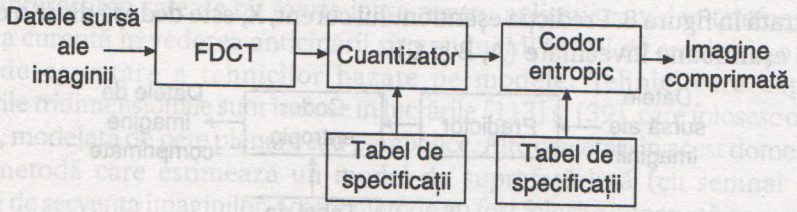


Figura 9. Codorul JPEG de tip BSC

Scopul etajului de cuantizare este de a prelua ieșirea etajului DCT și de a-i reduce entropia prin cuantizarea coeficienților. Acest etaj conduce la o pierdere de informație, dar asigură o reducere a ratei de bit din sistem. Prin ajustarea parametrilor din acest etaj, este posibil să se controleze rata de bit comprimată și calitatea imaginii la ieșire.

Codorul entropic preia coeficienții de cuantizare DCT de lungime fixă și produce un set de simboluri de canal de lungimi variabile. Această operație tinde să producă o compresie a fluxului de date, a cărui rată este cât se poate de apropiată de entropia coeficienților DCT cuantizați.

Transformata Cosinus Discretă Directă: Primul pas în procesul de compresie este extragerea blocurilor 8×8 din semnalul de intrare. Dacă intrarea are câteva canale de date, de exemplu RGB sau date CCIR601[NT10], fiecare canal de date este tratat identic. Dacă dimensiunile datelor de intrare nu sunt un multiplu de 8, intrarea este completată astfel încât să devină un multiplu întreg de blocuri 8×8. Se definește transformata cosinus discretă bidimensională conform ecuației (21) unde $X(u,v)$ este reprezentarea în frecvență a blocului de date de intrare $x(i,j)$. Pentru un bloc de intrare 8×8, $N = 8$ în ecuația (21) și $X(0,0)$ reprezintă termenul DC al blocului care este media celor 64 de eșantioane de intrare.

Cuantizarea: Cei 64 de coeficienți de la ieșirea transformatei DCT directe sunt apoi cuantizați uniform. Dimensiunea pasului de cuantizare, care este folosit pentru fiecare coeficient se găsește într-un tabel de cuantizare care trebuie specificat de către aplicație ca o intrare a codorului. Elementele tabelului de cuantizare pot lua valori întregi între 1 și 255.

[NT10] Recomandarea Nr. 601 a CCIR oferă un model de eșantionare, cuantizare, codare și împachetare a datelor pentru televiziunea digitală. A se vedea „Televiziunea digitală", editura Teora 1995.

Procesul de cuantizare se defineşte ca divizarea fiecărui coeficient DCT cu intrarea lui corespunzătoare din tabelul de cuantizare, urmată de rotunjirea la cea mai apropiată valoare întreagă.

$$X_Q(u,v) = \text{IntegerRound}\left(\frac{X(u,v)}{Q(u,v)}\right)$$

unde $Q(u,v)$ este tabelul de cuantizare.

În decodor, se realizează operaţia inversă, care furnizează decodorului intrările potrivite pentru transformata DCT inversă.

$$X_{Q'}(u,v) = X_Q(u,v) * Q(u,v)$$

Din această discuţie este clar că tabelul de cuantizare este parte din informaţia care trebuie transmisă de la codor la decodor.

Dacă o intrare a tabelului de cuantizare este mai mare decât unitatea, apar pierderi de informaţie. Aceste pierderi provoacă o reducere mare a ratei de date în codec. Matricea este astfel aleasă încât să asigure compromisul între eficienţa compresiei şi calitatea subiectivă a imaginii. În standard se prezintă un set de tabele de cuantizare pentru formatul de date CCIR601. Aceste tabele sunt deduse din experimente psiho-vizuale folosind proceduri date în lucrarea [92] (tabelele 2 şi 3).

Codarea entropiei: Înainte de codarea entropiei, se aplică o procesare suplimentară a coeficientului DC, de nivel continuu, X(0,0). Aşa cum s-a arătat mai sus, coeficientul DC reprezintă media valorilor blocului de intrare. Aceasta conduce la un grad mare de corelare între coeficienţii DC ai blocurilor transformate învecinate. În vederea exploatării acestei corelaţii, se calculează diferenţa între coeficienţii cuantizaţi DC ai blocurilor învecinate. Această valoare este folosită în etapele următoare în locul coeficienţilor cuantizaţi. La frecvenţe mari există o corelaţie mică între blocuri, de aceea nu se aplică nici o procesare ulterioară a coeficienţilor AC.

Tabelul 2. Tabel de cuantizare a canalului de luminanţă pentru date CCIR601

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Tabelul 3. Tabel de cuantizare a canalelor de crominanţă pentru date CCIR601

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Întrucât datele de imagine tind să aibă un spectru de frecvențe joase, coeficienții DCT cuantizați, diferiți de zero, se vor aduna în zona frecvențelor joase. Aceasta înseamnă, de asemenea, că coeficienții de frecvențe înalte vor tinde să se apropie de zero. Utilizând această proprietate se poate obține o mai bună compresie (fără pierderi). Dacă datele cuantizate ale coeficienților sunt reordonate folosind scanarea în zig-zag ca în figura 10, coeficienții nenuli vor tinde să fie transmiși primii. De asemenea, va fi un lung șir de coeficienți nuli la sfârșitul scanării. Folosind un marker de sfârșit de bloc, se elimină necesitatea de a transmite aceste zerouri de frecvențe înalte.

În acest punct, sunt puține valori nenule și mulți coeficienți DCT nuli. Codorul de entropie codează aceste date cât se poate de eficient. În BSC (*Baseline Sequential Codec*), acest proces apare în două trepte. Coeficienții DCT sunt transferați într-o reprezentare intermediară secvențială de simboluri, apoi cuvinte de cod sunt atribuite acestor simboluri.

În reprezentarea intermediară, fiecare coeficient AC este reprezentat de valoarea lui și numărul de valori nule consecutive care îl preced. Fiecare din aceste combinații de șiruri nule și valori nenule este reprezentat de o pereche de simboluri:

Simbol1: (ȘIR_UNIFORM, DIMENSIUNE)[NT11]

Simbol2: (AMPLITUDINE)

Simbol1 reprezintă două informații: ȘIR_UNIFORM, care este numărul de coeficienți nuli consecutivi, care preced coeficientul nenul, și DIMENSIUNE care este numărul de biți necesari pentru a coda amplitudinea. *Simbol2* reprezintă o singură informație: AMPLITUDINE, care este amplitudinea coeficientului cuantizat nenul. Aceste simboluri sunt codate utilizând diferite tabele Huffman. Primul tabel generează un cod de lungime variabilă, care reprezintă perechea (ȘIR_UNIFORM, DIMENSIUNE), în timp ce al doilea tabel codează un întreg de lungime variabilă care codează AMPLITUDINEA, iar interpretarea acestuia este definită de partea DIMENSIUNE din *simbol1*.

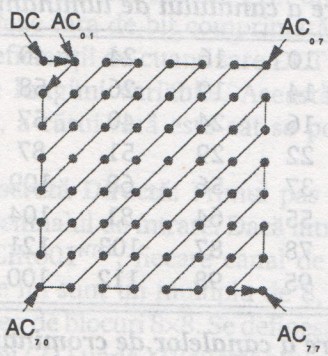


Figura 10. Scanarea în zig-zag folosită la codarea entropică

Alte moduri de operare: BSC este cel mai simplu dintre codecurile cu pierderi. Standardul complet definește câteva alte posibile moduri de operare, care pot fi utile în anumite aplicații. În toate cazurile, codorul entropic poate fi implementat fie ca un codor aritmetic, fie ca un codor Huffman.

[NT11] În engleză, symbol1 = (RUNLENGTH, SIZE)

Modul Progresiv de operare reține transformarea directă și etapele de cuantizare din modul de operare BSC. Diferența cheie este aceea că în loc să se trimită toți coeficienții cuantizați nenuli pentru fiecare bloc 8×8 în același timp, transmiterea lor este distribuită de-a lungul câtorva treceri prin date. Prima parcurgere a datelor produce o versiune a imaginii, de slabă calitate, însă recognoscibilă față de transmisia unui volum atât de mic de date. Următoarele parcurgeri ale datelor perfecționează această imagine până când, după terminarea ultimei parcurgeri, se obține calitatea finală a imaginii complete.

Se definesc două modalități de a segmenta datele de-a lungul acestor parcurgeri. Prima se numește „selecție spectrală". În acest caz, „benzi" de coeficienți de frecvențe crescătoare sunt trimise la fiecare parcurgere a datelor. De exemplu, prima trecere ar putea să transmită numai coeficienții DC ai fiecărui bloc. Trecerile următoare ar putea să transmită un număr de coeficienți de frecvență mai înaltă. Acest proces continuă până când toți coeficienții nenuli au fost transmiși.

O altă modalitate are la bază următoarea observație: coeficienții nenuli nu trebuie transmiși cu acuratețea maximă la o singură parcurgere a datelor. La o primă trecere se transmit cei mai semnificativi *N* biți din fiecare coeficient. La scanări succesive se transmit progresiv biții mai puțin semnificativi. Această abordare este cunoscută ca „aproximări succesive".

Ultima modalitate de operare este cea a codării ierarhice. Codoarele ierarhice operează pe o reprezentare piramidală a imaginii de intrare. Fiecare nivel al piramidei este o imagine care este jumătate din dimensiunea nivelului următor. Procedura de codare pentru acest mod poate fi sintetizată astfel:

1. Filtrarea și subeșantionarea imaginii cu un multiplu de 2 pe fiecare dimensiune[NT12].
2. Codarea acestui pachet folosind oricare dintre modurile de operare definite anterior.
3. Decodarea pachetului, supra-eșantionarea cu multiplul de 2 pe fiecare dimensiune.
4. Folosirea acestei imagini ca predictor. Se codează diferența între acest predictor și imaginea originală la această rezoluție, folosind oricare din modurile definite înainte.
5. Se repetă pașii 3 și 4 până când se obține codarea completă a imaginii la rezoluția dorită.

O aplicație interesantă a JPEG este codarea video bazată pe codarea JPEG intra-cadre a cadrelor în secvență. Avantajul acestei metode așa numite JPEG de Mișcare include o implementare de complexitate redusă, calitate a imaginii predictibilă și flexibilitate din punct de vedere al dimensiunilor și formatului imaginii. JPEG de Mișcare este folosită curent în aplicații ca editarea video neliniară și este, de asemenea, propusă pentru camcordere video digitale.

Un exemplu de compresie JPEG apare în partea stângă a figurii 14.

5.2. *Standardul de video teleconferințe H.261*

Acest standard, cunoscut și *ca p*64*, a fost dezvoltat cu scopul de a furniza un serviciu video standardizat pentru rețeaua ISDN (*Integrated Service Digital Network*). El definește un codec optimizat pentru aplicații de videofon și de video teleconferințe care poate opera la un set fix de rate de bit. Plaja sa de operare pleacă de la 64 kbps și merge până la 1920 kbps, în multipli ai ratei tipice pentru un canal ISDN de 64 kbps

[NT12] De exemplu se iau eșantioanele din 8 în 8 (decimare)

(p*64 unde p=1, 2, ..., 30). A fost aprobat în decembrie 1990[20] ca un standard internațional. În plus, o versiune ușor modificată a fost adoptată pentru a fi utilizată în America de Nord.[6]

Dat fiind că aplicația dorită era interactivă, algoritmul de compresie video trebuia să opereze cu întârziere minimă. Pentru valori mici ale lui p, el este potrivit pentru aplicații videofon. Dacă este disponibilă o bandă mai mare, $p \geq 6$, există o disponibilitate suficientă de rată de bit pentru aplicații de tip videoconferință. Informații suplimentare față de documentele standardelor pot fi găsite în lucrarea [88].

Datele de intrare pot fi definite fie în format CIF, fie QCIF, așa cum s-a definit în capitolul 2. Sintaxa fluxului de biți este divizată într-un set de niveluri care descriu structura fiecărui cadru. Acestea sunt imagine (Picture), grup de blocuri (GOB), macro-blocuri (MB) și blocuri (B). Un macro-bloc constă în patru blocuri de 8×8 pixeli de luminanță și două blocuri de 8×8 de crominanță, câte unul pentru C_r și C_b. Un GOB este compus din 3×11 macro-blocuri. O imagine QCIF conține 3 GOB în timp ce o imagine CIF conține 12 din fiecare canal de crominanță.

Algorimul de compresie este bazat pe modelul (figura 11) transformării MC-DCT (*Motion Compensated-DCT*). Punctul de plecare pentru abordarea de bază a MC-DCT este folosirea unui filtru buclă (FILTRU). Acest filtru poate fi deschis sau închis și servește la atenuarea zgomotelor de frecvență înaltă în bucla de reacție. Deoarece codecul este definit pentru operare la rate de bit constante, ieșirea codorului trece printr-un buffer (separator – etaj de stocare temporară) înainte de transmisie. Starea bufferului este monitorizată și folosită pentru a deduce mărimea pasului de cuantizare. Pe măsură ce bufferul se umple, mărimea pasului crește, ceea ce duce la mai puțini biți și la creșterea distorsiunilor de cuantizare. Dacă bufferul este golit prea repede, pasul de cuantizare este micșorat, ceea ce duce la mai mulți biți, dar la distorsiuni mai mici.

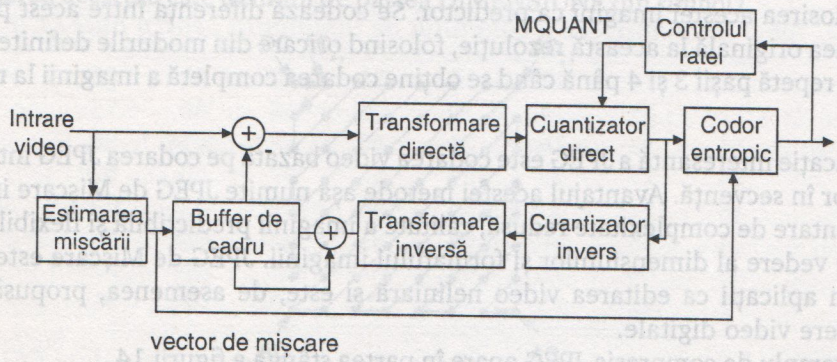


Figura 11. Structura unui codor bazat pe MC-DCT

Sintaxa fluxului de biți este ilustrată în figura 12. Toate nivelurile conțin un cod de start, folosit în scopul sincronizării, care este urmat de alte informații necesare. Pe nivelul Imagine, informația constă din formatul video (CIF/QCIF) și referința temporală (numărul cadrului). Ar trebui notat că referințele temporale nu trebuie să fie consecutive. Aceasta permite codorului să nu codeze cadrele de intrare dacă nu are suficient loc în buffer.

Nivelul GOP constă dintr-un header (titlu) și din blocul de date Macro-bloc. Headerul conține poziția de start a GOM în cadrul Imaginii și dimensiunea pasului de cuantizare.

Nivelul Macro-bloc constă dintr-un header MB și din datele Blocurilor. Headerul MB conține adresele macro-blocurilor în cadrul GOB și flaguri care indică dacă MB este codat inter-cadre sau intra-cadru, cu sau fără compensarea mișcării și dacă bucla filtrului este deschisă sau închisă. Dacă este cazul, vor urma datele de Bloc. Dacă compensarea mișcării este suficient de precisă, se poate să nu fie necesară transmiterea coeficienților DCT. În plus, nu este necesar să fie transmise toate MB dintr-un GOB.

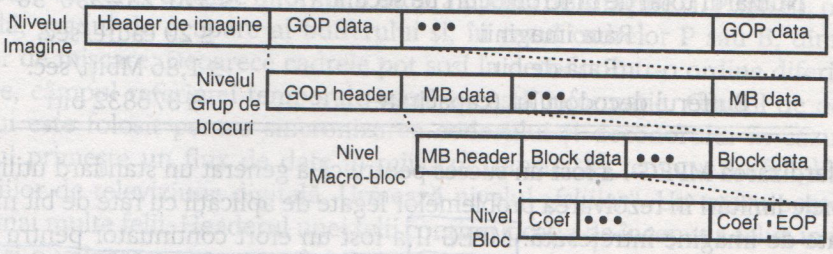


Figura 12. Fluxul de biți pentru codorul H.261

Nivelul Bloc conține datele coeficienților DCT codate cu lungime variabilă. Ar trebui notat că, pentru a diminua efectele erorilor de transmisie, fiecare Macro-bloc trebuie transmis uneori în mod intra-cadru.

5.3. *Standardul de compresie Video MPEG*

În ultimii șapte ani a fost dezvoltat un cadru pentru standardizarea stocării și transmisiei datelor audio-vizuale. Primul dintre acestea, MPEG-I sau ISO11172, a fost inițial focalizat pe codarea imaginilor dinamice și a semnalelor audio asociate pentru stocarea digitală pentru mijloacele de comunicare (DSM = Digital Storage Media). Cu timpul, s-a observat că tehnica explorată a devenit de utilitate generică, așa că DSM a fost înlocuită cu termenul „generic".

În MPEG-I, dimensiunea maximă a imaginii este 4096×4096 de pixeli și standardul suportă câteva viteze între 24 și 60 de cadre pe secundă. Utilizarea cea mai tipică a MPEG-I este în compresia video SIF pentru distribuția de filme pe disc compact cu o rată de codare video de 1,1 Mbps. Calitatea imaginii în această aplicație este dependentă de scenă și este în general la nivelul oferit de video-playerele VHS. Pentru a permite implementarea la costuri reduse a decodoarelor pentru aplicații ca filmele pe CD, un set limitativ de parametrii a fost definit și s-au impus limite maxime pentru valorile unor parametri (tabelul 4). Acești parametri sunt direct legați de complexitatea hardware a decodorului. Dimensiunea imaginii și restricțiile asupra interpolării cadrelor se leagă direct de volumul de memorie al bufferului de cadre necesar. Restricțiile macroblocurilor limitează superior lățimea de bandă cerută memoriei și limita ratei de bit în urma compresiei este condiționată direct de capacitatea necesară în decodorul de entropie. Cu excepția acestor limitări, nu sunt alte restricții ale formatului materialului sursă.

Tabelul 4. MPEG-I oferă un set limitativ de parametri, care vizează aplicațiile de filme pe CD-ROM

Setul de parametri limitativi MPEG	
Dimensiunea orizontală	≤ 720 pixeli
Dimensiunea verticală	≤ 576 pixeli
Numărul total de macroblocuri pe imagine	≤ 396
Numărul total de macroblocuri pe secundă	≤ 396*25=330*30
Rata imaginii	≤ 20 cadre/sec.
Rata de bit	≤ 1,86 Mbiți/sec.
Bufferul decodorului (capacitate)	≤ 376832 biți

Standardizarea MPEG-I a fost un succes pentru că a generat un standard utilizabil. Dar a avut unele limitări în rezolvarea problemelor legate de aplicații cu rate de bit mai mari și cu formate de imagine întrețesută. MPEG-II a fost un efort continuator pentru a rezolva aceste probleme. Inițial, urma să aibă loc un alt efort de standardizare (MPEG-III), care s-ar fi adresat televiziunii de înaltă definiție (HDTV). Acest lucru a fost anulat când s-a constatat că definiția asigurată de MPEG-II are flexibilitatea de a satisface și aceste aplicații.

Caracteristicile MPEG-I. Standardul video MPEG-I oferă un instrument orientat către compresia progresivă video pentru stocarea multimedia. Toate standardele MPEG sunt bazate pe cadrele MC-DCT descrise în capitolul 4.1. Trăsăturile unice ale standardului Video MPEG-I vor fi prezentate în acest capitol. MPEG-I oferă o sintaxă stratificată a fluxului de date, care permite o mare flexibilitate atât în domeniul aplicațiilor cât și în domeniul optimizării codoarelor. Nivelul de sus al sintaxei video, *nivelul sistem*, oferă posibilitatea de a specifica dimensiunile imaginii, raportul de aspect, rata cadrelor de date, rata de bit în urma compresiei și dimensiunea bufferului decodorului de compresie. Pentru MPEG-I, imaginea trebuie să fie mai mică de 4096×4096, rata de bit în urma compresiei trebuie să fie mai mică de 104 Mbps, iar bufferul cadrului comprimat are o dimensiune maximă a memoriei de 16384 kb.

Cuantizatorii folosiți de MPEG-I sunt uniformi, pentru date intra-codate și uniformi, cu o zonă moartă, pentru datele codate predictiv. Aceiași cuantizatori sunt folosiți atât pentru canalul de luminanță cât și pentru cele de crominanță, dar sunt furnizate matrice diferite pentru cuantizările inter- și intra-. Un set de matrice sunt date implicit în standard și acestea vor fi folosite de decodor implicit, dacă nu se transmite o altă matrice pe nivelul sistem.

Următorul nivel al sintaxei se numește nivelul GOP (*Group Of Pictures* = grupul de imagini). Structura grupului de imagini a fost înzestrată să permită încorporarea unor puncte de acces aleator în fluxul de date comprimat și abilitatea de a realiza „funcții VCR" cum ar fi derularea rapidă înainte și înapoi. Primul cadru din fiecare GOP trebuie să fie codat în întregime intra-cadru. Nu există nici o restricție de lungime a GOP. El continuă până când în fluxul de date apare un cod de start al unui alt GOP.

Un GOP poate conține trei tipuri de cadre. Acestea sunt (I) intra-codate, (P) codate predictiv nerecursiv și (B) codate bidirecțional predictiv. Cadrele intra-codate pot fi decodate independent de orice alt cadru și oferă cea mai mică rată de compresie. Cadrele codate predictiv depind de datele (vectori de mișcare) din cadrele precedente și, în cele mai multe cazuri, oferă o rată de compresie mai bună decât cadrele I. Cadrele B sunt prezise atât din cadrele precedente cât și din cele următoare și oferă cele mai înalte rate

de compresie. Numărul cadrelor B consecutive permise este limitat, deoarece acesta influențează complexitatea implementării codorului. Cele mai multe implementări impun o limită de doi. Ar trebui, de asemenea, notat că aceste dependențe ale predicției afectează ordinea de transmitere a datelor. Decodorul trebuie să aibă toate datele asupra cadrelor de care depinde decodarea cadrului curent, înainte de a decoda cadrul curent. Aceasta înseamnă că ordinea de transmisie diferă de ordinea de redare.

Al treilea nivel al fluxului de date, *nivelul de imagine*, este folosit pentru a descrie un singur cadru video. El conține informații asupra tipului de imagine (I, P sau B), o referință temporală, gradul de ocupare al bufferului și, în cazul cadrelor P sau B, dimensiunea vectorilor de mișcare. Deoarece cadrele pot sosi la decodor într-o ordine diferită de cea de redare, câmpul referinței temporale oferă această informație. Gradul de ocupare al bufferului este folosit pentru sincronizarea codorului și decodorului în cazul în care decodorul primește un flux de date în mijlocul transmisiei, cum se întâmplă în cazul transmisiilor de televiziune digitală. Urmează nivelul *„feliilor"*. Un cadru poate conține una sau mai multe felii. Headerul unei felii conține locația de început a feliei în interiorul cadrului și o valoare a parametrului de scală al cuantizatorului (MQUANT). MQUANT poate lua valori în plaja 1-31. El este multiplicat de matricea cuantizatorului și este folosit la stabilirea ratei de bit la ieșire. Informația de poziție a feliei este folosită la resincronizarea decodorului, când sunt erori de canal. Apoi vin nivelurile *macrobloc* și *bloc*. Aceste niveluri transmit vectorii de mișcare și coeficienții DCT într-o manieră similară cu H.261.

Caracteristicile MPEG-II. MPEG-I a fost optimizat pentru compresia materialelor sursă progresive la o rezoluție aproximativ egală cu VHS. MPEG-II a fost inițiat în dorința de a extinde acest standard pentru a satisface calitatea surselor radiodifuzate întrețesute și, mai târziu, materialele de televiziune de înaltă definiție.

Pentru a ajunge la acest punct, au fost adăugate câteva noi caracteristici. Pentru a da producătorilor posibilitatea de a oferi soluții ieftine pe diferite piețe, a fost definit un set de *Niveluri* și de *Profiluri* (tabelul 5). În general, profilurile de nivel mai înalt cer decodorului să suporte mai multe funcții, în timp ce nivelurile mai înalte cer decodorului să suporte rezoluții mai mari și rate de compresie a datelor mai ridicate.

Tabelul 5. Standardul MPEG-II are flexibilitatea de a suporta o mare varietate de formate de intrare și rate de date, prin definirea nivelurilor și profilurilor

Profil		Nivel scăzut		
		Rezoluție max.	Rată de cadre max.	Rata de bit max.
Nescalabil	4:2:0 principal	352x288	30	4
Scalabil	4:2:0 principal +	352x288	30	4 (toate nivelurile) 3 (nivelul de bază)

Profil		Nivel principal				
		Rezoluție Max.	Rata de cadre max.	Rezoluție Min.	Rata de cadre min.	Rata de bit max.
Nescalabil	Simplu 4:2:0	720x576	30	N/A	N/A	15
	Principal 4:2:0	720x576	30	N/A	N/A	15
Scalabil	Principal + 4:2:0	720x576	30	N/A	N/A	15 (toate nivelurile) 10 (niv. bază)
	Următorul 4:2:2	720x576	30	352x288	30	20 (toate nivelurile) 15 (niv. bază +mijl.) 4 (niv. bază)

Tabelul 5. (continuare)

Profil		Nivel înalt – 1440				
		Rezoluție Max.	Rata de cadre max.	Rezoluție Min.	Rata de cadre min.	Rata de bit max.
Nescalabil	Princ. 4:2:0	1440x1152	60	N/A	N/A	60
Scalabil	Princ. + 4:2:0	1440x1152	60	720x576	30	60 (toate nivelurile) 40 (niv. bază+mijl) 15 (niv. bază)
	Următorul 4:2:2	1440x1152	60	720x576	60	80 (toate nivelurile) 60 (niv. bază +mijl.) 20 (niv. bază)

Profil		Nivel înalt				
		Rezoluție Max.	Rata de cadre Max.	Rezoluție Min.	Rata de cadre Min.	Rata de bit max.
Nescalabil	Princ. 4:2:0	1920x1152	60	N/A	N/A	80
Scalabil	Următorul 4:2:2	1920x1156	60	960x576	30	100 (toate nivelurile) 80 (niv. bază+mijl.) 25 (niv. bază)

Una dintre principalele diferențe de caracteristici între MPEG-I și MPEG-II constă în opțiunile pentru predicția de compensare a mișcării. Deoarece MPEG-II trebuie să se ocupe de materiale sursă întrețesute, el oferă posibilitatea de a prezice macroblocul curent din cadrul precedent, sau câmpul (semicadrul) precedent de aceeași fază, sau câmpul precedent de fază opusă. În plus, independent de modul de predicție, DCT și cuantizarea pot avea loc fie pe datele de câmp fie pe datele de cadru.

În vederea asigurării posibilității de codare ierarhică, se definesc două moduri scalare. Scalabilitatea *SNR* este similară cu „aproximațiile succesive" din JPEG. În primul nivel se transmite un subset de coeficienți DCT. Această imagine de calitate redusă este folosită ca predicție pentru o altă etapă de codare MC-DCT, care induce un flux de date mai mare. Aceste două fluxuri de date permit transmiterea unei imagini de calitate redusă utilizatorilor care au un canal de capacitate limitată și simultan transmiterea unei imagini de calitate mai bună, la aceeași rezoluție, utilizatorilor care au un canal de capacitate mai mare.

Celălalt mod de scalare este numit *scalabilitate spațială*. Acest mod este similar cu „codarea ierarhică" din JPEG. Diferența constă în faptul că pot fi numai două niveluri ierahice, iar scalarea poate fi ori spațială, ori temporală. Aceasta înseamnă că mărirea fluxului de date poate crește fie rezoluția spațială a fluxului de date de bază, fie să crească rata lui temporală. Un exemplu de scalabilitate temporală este cazul în care nivelul de bază este un material sursă întrețesut (ca în MPEG-I) în timp ce formatul mărit este progresiv. O posibilă utilizare a scalabilității spațiale este de a avea nivelul de bază în format CCIR601 și să se folosească nivelul mărit pentru a transmite versiunea HDTV.

Aplicația originală a MPEG-II a fost de a reproduce secvențe de film la rate de bit între 4 și 10 Mbps cu o rezoluție a imaginii corespunzătoare formatului CCIR601, care include sub-eșantionarea color pe direcția verticală. Standardul MPEG-II a fost, de asemenea, propus ca standard al Statelor Unite pentru transmisia HDTV cu rate de bit în jurul a 20 Mbps.

6. Pre- și postprocesarea pentru compresia imaginilor

La rate de bit reduse, calitatea imaginii începe să se degradeze, producând artefacte (componente perturbatoare) vizuale deranjante. Diferite tehnici de compresie produc artefacte diferite. Tehnicile de pre- și post-procesare au fost introduse pentru a atenua unele dintre distorsiunile tipice produse. Ideal, codorul ar trebui să conțină implicit orice pași de pre- și postprocesare, sau să fie proiectat astfel încât acești pași adiționali să nu fie necesari.

Preprocesarea. Cel mai comun pas de preprocesare este decimarea (reducerea) secvenței video în spațiu și în timp. Este uzual pentru mulți algoritmi de compresie, care lucrează la rate de bit mai mici de 384 kbps, de a scădea rezoluția spațială a imaginii de la SIF la QSIF (1/4 din SIF). Tipic, frecvența cadrelor este de 30 cadre/sec (c/s) și este scăzută la mai puțin de 10 c/s. [NT13] În mod ideal, cele mai eficiente tehnici de preprocesare necesită analiza algoritmului de compresie neliniară și dezvoltarea celui mai bun algoritm de preprocesare pentru datele video, bazat pe ceea ce va deveni imaginea după procesul de codare/ transmisie/ decodare. Aceasta este foarte dificil de analizat și foarte puține cercetări s-au focalizat pe algoritmi de preprocesare pentru algoritmi de compresie. Unele cercetări foarte interesante, bazate pe folosirea unor indicații perceptuale în cadrul unui sistem tradițional de codare, ar putea fi descrise ca tehnici de pre- și postprocesare. Acestea includ cercetări asupra precuantizării perceptuale.[156] Aceste tehnici funcționează în cadrul standardelor (JPEG și MPEG) oferind un câștig semnificativ de codare în pasul de precuantizare perceptuală. Altă lucrare de pre- și postprocesare pe baze perceptuale este inclusă în [194], care încorporează un model vizual perceptual ca parte a măsurării distorsiunilor într-o schemă de codare cu cuantizare vectorială. Etapa de preprocesare constă într-un model vizual și un filtru de preaccentuare, în timp ce etapa de postprocesare constă într-un filtru de dezaccentuare și modelarea vizuală inversă.

Alte tehnici care ar putea fi descrise ca tehnici de prefiltrare pentru compresia imaginii sunt acelea care identifică și analizează ariile critice ale secvenței video, pentru a aloca mai mulți biți acestor zone. S-au făcut multe cercetări pentru identificarea și segmentarea fețelor, atât pentru aplicațiile de compresie a imaginii cât și pentru alte aplicații, cum ar fi recunoașterea și sinteza feței. S-au făcut cercetări pentru localizarea feței și a elementelor faciale pe baza încadrării unui șablon în contururile semnalului de intrare.[163,68,31,198] Folosirea unei tehnici de localizare a feței pentru alocarea de biți într-un codor standard este descrisă în lucrarea [40].

Preprocesarea poate fi, de asemenea, utilizată ca o metodă pentru a încorpora noi funcții în tehnicile de compresie existente, care nu au fost anticipate atunci când au fost create aceste tehnici. Un exemplu în acest sens este aplicarea unei paradigme de codare perceptuală descrisă în capitolul 4.1, codorului standard de compresie cu pierderi pentru imagini statice JPEG.[156,157] Problema pe care o întâlnim când dorim să îmbunătățim un codec existent este că există deja definită o sintaxă a fluxului de date JPEG, cu care trebuie să păstrăm compatibilitatea. Provocarea constă în a găsi posibilitățile de utilizare mai eficientă a capacităților disponibile din cadrul sintaxei existente a fluxului de date. Așa cum am mai afirmat anterior, codarea perceptuală operează pe baza pragurilor

[NT13] Textul se referă la standardul NTSC american, unde frecvența cadrelor este 30 c/s (jumătate din frecvența rețelei electrice). În Europa, frecvența cadrelor este tipic 25 c/s.

JND (*Just Noticeable Difference* = diferențele abia observabile) pentru fiecare coeficient și apoi codarea acestor date astfel încât aceste praguri de distorsiuni să nu fie depășite.

Această problemă constă în a găsi un model de mascare vizuală, care să fie utilizat în mediul JPEG. De vreme ce JPEG permite utilizatorului să specifice tabele de cuantizare pentru fiecare imagine, ar trebui să fie posibil să se deducă unele „perceptual optimale". De fapt, în lucrarea [138] este prezentat un set de formule pentru determinarea matricelor de cuantizare optimă (atât pentru luminanță cât și pentru croma), date fiind dimensiunea imaginii, punctul de alb de referință și distanța de privire. Aceste formule au fost deduse în urma unui lung șir de experimente pentru a determina detectabilitatea fiecărei funcții DCT de bază. Din aceste experimente a fost dedus un model de predicție a pragurilor de detecție bazată numai pe condițiile de vizionare și pe proprietățile globale ale sistemului vizual (funcția de transfer de modulație).

Această cercetare oferă un început promițător pe calea spre un codor pe baze perceptuale. Slăbiciunea constă în faptul că se apelează numai la mascarea globală. Este bine cunoscut faptul că multe fenomene de mascare au loc la un nivel local. Acest lucru a făcut obiectul lucrării [188], care a extins această abordare pentru a determina un tabel de cuantizare bazat pe imagine, care să încorporeze nu numai condiții globale, ci să conteze și pe mascarea contrastului local. Se prezintă un algoritm iterativ, care determină un tabel de cuantizare bazat pe imagine, dependent de imagine, care oferă un anumit nivel specificat de distorsiuni de imagine. Această abordare este o îmbunătățire față de abordarea globală, dar încă nu permite adaptarea locală. Poate fi însă aplicată într-o manieră non-iterativă. Acest model de mascare, ca și celălalt, determină factorul de elevație a pragului. Aceasta înseamnă că local, valoarea JND pentru fiecare coeficient poate fi calculată ca o funcție de tabelul de cuantizare global și de valoarea coeficienților DCT ai blocului local.

Aceasta conduce la un set de praguri de mascare pentru fiecare bloc al unei imagini de intrare, care oferă un tabel de cuantizare unic perceptual, optimal pentru firecare bloc. Din păcate, JPEG permite numai un singur tabel de cuantizare pentru fiecare imagine. Prin urmare, o problemă care trebuie rezolvată este cum să folosim această informație locală în cadrul standardului JPEG. Dacă examinați ecuația cuantizării nerecursive din capitolul 5.1, este clar că toți coeficienții care au valori mai mici decât intrarea lor corespunzătoare din tabelul de cuantizare, vor fi cuantizați la valoarea zero. Această observație este cheia încorporării cuantizării locale adaptive în JPEG.

De vreme ce coeficienții cuantizați cu valoare zero constituie un număr valid al fluxului de date JPEG, codorul perceptual va identifica pe care coeficienți îi poate aduce la zero păstrând calitatea subiectivă a imaginii codate. Aceasta va menține compatibilitatea cu specificația fluxului JPEG în timp ce va reduce rata de bit necesară pentru a coda imaginea. Figura 13 ilustrează structura unui astfel de codor. Transformarea nerecursivă este identică cu BSC JPEG. În acest punct, coeficienții DCT sunt intrări în modelul perceptual care generează tabelul de cuantizare dependent de date al acelui bloc. Acest tabel și coeficienții bruți DCT sunt intrări în „pre-cuantizator". Scopul acestui model este să aducă la zero coeficienții care au amplitudinea mai mică decât intrarea corespunzătoare a tabelului de cuantizare pentru acel bloc și să păstreze neschimbați ceilalți coeficienți.

Acest proces poate fi privit ca o funcție *poartă* dependentă de date, care permite ca numai coeficienții importanți perceptual să fie aplicați cuantizatorului standard JPEG. În final, acești coeficienți precuantizați sunt cuantizați și codați entropic ca în standardul JPEG.

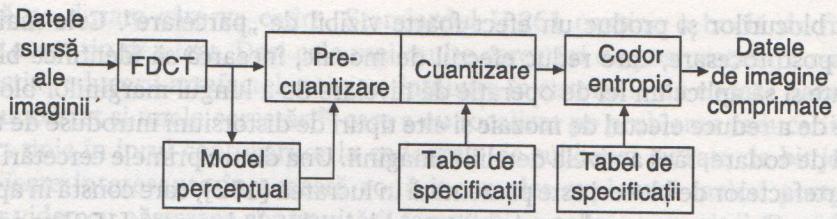


Figura 13. Diagrama bloc pentru codorul P×JPEG

Rezultatele acestui proces sunt ilustrate în figura 14. Imaginea din stânga a fost comprimată folosind BSC astfel încât a fost perceptual fără pierderi la o distanță de privire egală cu de șase ori înălțimea ecranului. Imaginea din dreapta a fost comprimată la același nivel de calitate folosind abordarea precuantizării perceptuale. Folosind codarea perceptuală s-a obținut o economie de 26% la rata de bit.

Postprocesarea. Postprocesarea este o problemă mult mai bine definită: după etajele de codare / decodare, reduce artefactele vizibile introduse de algoritmul de compresie. Deși postprocesarea este limitată datorită faptului că nu poate reface sau recupera datele care au fost eliminate de algoritmul de compresie, este mai ușor de urmărit decât preprocesarea.

Figura 14. BSC JPEG la o rată de 1,23 biți / eșantion (stânga); P×JPEG la aceeași calitate, la o rată de bit de 0,91 biți / eșantion (dreapta)

Problema cel mai des întâlnită în postprocesarea compresiei video este eliminarea artefactelor de „mozaic" introduse de transformarea DCT bazată pe blocuri. Aceasta este o problemă importantă deoarece transformarea bloc-DCT este o parte integrantă a tuturor standardelor de codare a imaginilor statice și video și este, de asemenea, larg utilizată în algoritmii de căutare. Un tip de distorsiune mozaic este introdus deoarece fiecare bloc este codat *independent*. Acest tip de distorsiune nu este legat numai de DCT ci fiecare tip de schemă de compresie care se bazează pe procesarea independentă a blocurilor poate introduce distorsiuni similare.[i] Distorsiunile de mozaic apar ca discontinuități de-a lungul

[i] Ideea transformatei ortogonale (LOT) descrisă în subcapitolul 3.2 este destinată tocmai eliminării acestor distorsiuni. O generalizare mai mare, codarea subbenzilor, evită, de asemenea, problema distorsiunilor de mozaic.

marginilor blocurilor şi produc un efect foarte vizibil de „parcelare". Cele mai multe tehnici de postprocesare, care reduc efectul de mozaic, încearcă să identifice blocurile distorsionate şi să aplice un fel de operaţie de nivelare de-a lungul marginilor blocurilor. Trucul este de a reduce efectul de mozaic şi álte tipuri de distorsiuni introduse de tehnica particulară de codare, fără a nivela detaliile imaginii. Una dintre primele cercetări asupra reducerii artefactelor de blocuri este prezentată în lucrarea [152], care constă în aplicarea unui filtru trece jos pentru a reduce vizibilitatea blocurilor. În lucrarea [150] se aplică un filtru trece jos paralel cu marginile imaginii, pentru a nivela mozaicul o dată cu păstrarea claritătii contururilor. Prin formularea problemei postprocesării ca o problemă de refacere, se pot aplica multe alte tehnici eficiente şi interesante pentru a reface imaginea.[196,197,93] Multe dintre metodele de restaurare a imagini sunt bazate pe teoria proiecţiei pe un set convex (POCS = *Projection Onto Convex Set*). Această idee se bazează pe aplicarea informaţiilor *a priori* despre semnal, pentru a-l reface. Atâta vreme cât constrângerile *a priori* pot fi exprimate ca seturi de soluţii convexe, putem proiecta iterativ pe fiecare din seturile convexe, către o soluţie fezabilă, presupunând că intersecţia seturilor convexe, nu este nulă. O constrângere puternică, ce poate fi utilizată într-o schemă de postprocesare bazată pe POCS, este *constrângerea cuantizării*. Constrângerea forţează valorile pixelilor să rămână între limitele maximă şi minimă definite de nivelurile de cuantizare şi de mărimea pasului. Nivelarea este altă constrângere uzuală care poate fi aplicată în multe feluri. Pe măsură ce constrângerea devine mai legată de sistemul de codare, postprocesarea poate deveni mai eficientă, dar mai puţin utilă ca o unealtă de postprocesare de uz general. Ideal, în loc să folosim postprocesarea pentru a atenua artefactele de blocuri, am prefera să folosim scheme de compresie care să nu introducă astfel de distorsiuni, adică un codor care nu se bazează pe blocuri. Oricum, toate standardele curente de compresie a imaginii se bazează pe transformările DCT de blocuri.

Un alt tip de distorsiune caracteristic codoarelor MC/DCT este variaţia temporală a zgomotului de frecvenţă înaltă. În semnalele video dinamice, acest tip de distorsiune este cunoscută sub numele de „purici" (în engleză = „mosquito noise"), o ţesătură de frecvenţe înalte, aleatoare, în permanentă schimbare. S-au făcut foarte puţine eforturi pentru a reduce acest tip de distorsiuni. O excepţie o constituie lucrarea [8], unde se face referire la ambele tipuri de distorsiuni, „mozaic" şi „purici". Artefactele de bloc se identifică folosind o tehnică foarte simplă, de a număra coeficienţii nenuli din fiecare bloc. Mozaicul este ameliorat prin aplicarea unui filtru trece jos de-a lungul marginilor blocurilor marcate. Zgomotul de purici este diminuat prin folosirea unui detector de fronturi[89] pentru a determina o hartă a fronturilor, urmată de o nivelare a hărţii fronturilor bazată pe păstrarea numai a fronturilor care apar pe direcţiile orizontală, verticală sau în diagonală. Aceasta elimină zonele zgomotoase care sunt identificate ca fronturi false de detectorul de fronturi. Un filtru de nivelare este aplicat datelor „fără fronturi", unde poate să existe sau nu zgomot de frecvenţă înaltă. Această idee se bazează pe premisa că la rate de bit foarte scăzute, detaliile de imagine de înaltă frecvenţă nu sunt conservate foarte bine şi deci nivelarea datelor de imagine nu va degrada foarte tare calitatea imaginii. Această tehnică asigură rezultate bune codoarelor bazate pe MC/DCT la rate de bit foarte scăzute. Alt tip de distorsiuni pe care le găsim în urma compresiei bazate pe MC este efectul de „fereastră murdară". Acest tip de distorsiuni se datorează unei predicţii slabe a mişcării compensate, unde distorsiunea

persistă pe durata câtorva cadre. Standardul H.261 conține o buclă de filtrare care reduce acest tip de efect. Deși cele mai multe cercetări de postprocesare s-au efectuat în direcția reducerii artefactelor tipice, întâlnite în sistemele de codare bazate pe MC/DCT, au existat și unele cercetări[89] care s-au focalizat pe problema reducerii „riplului", artefact tipic în jurul contururilor la codoarele de subbenzi cu rate de bit joase.

Subiecte interesante în această zonă incomplet explorată includ postprocesarea datelor video cu păstrarea continuității temporale a calității imaginii. Dacă între cadre calitatea perceptuală oscilează, aceasta poate fi mai observabilă decât distorsiunile temporale de uniformitate a imaginii. Postprocesarea cu un model perceptual mai explicit ar putea, de asemenea, să aducă îmbunătățiri. Pe scurt, deși unele pre- și postfiltrări s-au dovedit utile, în particular în cadrul standardelor curente de codare, codorul ideal ar trebui să conțină implicit un preprocesor și ar trebui să nu aibă nevoie de nici un etaj de postprocesare.

7. Probleme de complexitate

În plus față de asigurarea unei calități perceptuale înalte, codecurile de imagine și video trebuie, de asemenea, să fie competitive din punct de vedere al complexității implementării. Cum codecurile de imagine și video sunt incluse în produsele de larg consum, costurile de implementare și consumul de putere devin criterii importante în selecția unui algoritm particular de compresie.[62] Dacă toți ceilalți factori sunt egali, un algoritm care oferă o ușoară creștere a calității imaginii cu prețul unor creșteri masive ale costurilor de implementare, nu va fi folosit. Acest comportament rigid se datorează sensibilității extreme față de preț a electronicii de consum.

Complexitatea implementării poate fi caracterizată de două componente: memoria și logica combinațională. Memoria se referă la numărul de cuvinte de memorie necesare pentru a stoca datele necesare implementării algoritmului. Logica combinațională oferă o estimare a numărului și tipurilor de operații aritmetice și logice care trebuie efectuate.

Pentru implementarea hardware subcontractată, numărul și tipurile de unități algoritmice (de exemplu DCT, de/codarea Huffman, estimarea mișcării) sunt tot ceea ce este necesar. Costul exact al fiecărei unități, măsurat în termeni de număr de porți depinde de inventivitatea proiectantului hardware (de circuite integrate). Pentru implementări numai software, logica combinațională poate fi măsurată în MIPS (milioane de instrucțiuni pe secundă) și în dimensiunea memoriei principale.

Logica combinațională poate fi măsurată în câteva moduri. Cel mai uzual este să se folosească MIPS pentru cazul cel mai defavorabil, care oferă o estimare a tipului de procesor necesar pentru implementarea software. Acest lucru este extrem de dificil de determinat fără a face o implementare atentă pentru un procesor particular, datorită numărului diferit de unități algoritmice din aceste sisteme și a numărului de posibilități de implementări pentru fiecare.

O altă cale de a măsura complexitatea este de a folosi o măsură de *macroblocuri pe secundă*. De vreme ce toți algoritmii de compresie standard operează pe baza macroblocurilor, această măsură oferă mijloacele de comparație a numărului de unități de bază care trebuie procesate în unitatea de timp. Dacă se cunoaște complexitatea procesării unui macrobloc, această măsură oferă un mijloc rapid de a compara sisteme concurente.

Un sumar de estimări ale complexităților pentru câțiva algoritmi standard este prezentat în tabelul 6. Intrările MIPS sunt estimări în cazurile cele mai defavorabile și nu includ operațiile de pre- și postprocesare cum ar fi conversia în spațiul culorilor și supra- sau subeșantionarea semnalului croma. Estimarea codorului pentru codecurile video presupune că se folosesc trei pași de investigare pentru estimarea mișcării. Investigarea completă pentru estimarea mișcării ar genera o complexitate mult mai mare.

JPEG necesită buffer pentru numai un rând de date macrobloc la un moment dat de timp, astfel încât are necesități modeste de memorie. De vreme ce H.261 este un codor compensat de mișcare, are nevoie de un cadru de memorie pentru a păstra cadrul de referință. MPEG permite atât predicție nerecursivă, ceea ce necesită un singur cadru de referință, cât și predicție bidirecțională, care necesită două cadre de referință. Aceasta implică necesitatea codecurilor MPEG de a avea suficientă memorie pentru a stoca două cadre de date necomprimate. În plus, aceste codoare necesită o mică memorie pentru bufferele de canal.

Tabelul 6. Acest tabel oferă estimări ale complexității implementării câtorva standarde de compresie de imagine și video

Algoritmul	MIPS		Macroblo-curi / sec.	Memorie (kB)
	Codorul	Decodorul		
JPEG (640×480 pixeli 1 imagine / sec)	6	3	1200	17
H.261 (352×288 pixeli, 15 cadre / sec.)	500	40	5940	150
MPEG-I (352×240 pixeli, 30 cadre / sec.)	650	50	9900	125
MPEG-II (720×480 pixeli, 30 cadre / sec.)	5000	100	40500	1015
HDTV (1280×720 pixeli, 60 cadre / sec.)	20000	500	216000	2700

8. Direcții viitoare

Avansul tehnologiei compresiei semnalelor în ultimele câteva decade a fost foarte impresionant. Se așteaptă o limitare a posibilităților de compresie datorită entropiei inerente a semnalelor cu care lucrăm și datorită limitărilor impuse de sistemele practice. Oricum, gândim că sunt multe domenii de cercetare rămase pentru a fi explorate. Un domeniu în care abia am zgâriat suprafața în înțelegerea și aplicarea compresise video este sistemul vizual uman și fenomenele psihovizuale. Se poate câștiga mult prin înțelegerea sistemului vizual uman și aplicarea acestor cunoștințe în sistemele de compresie a semnalelor, în special la rate de bit reduse. Simțim că se pot realiza câștiguri semnificative în această direcție.

Continua extindere a aplicațiilor multimedia și constrângerile de conectivitate în rețele creează noi cerințe în sistemele de compresie a imaginilor. Au existat unele lucrări care au investigat scheme de compresie a imaginii în contextul rețelelor.[107,106,184,57,178,173,79,80] Trebuie să reexaminăm tehnicile curente de compresie și să inventăm noi metodologii special focalizate pe problemele de codare sursă / canal, care să corespundă cerințelor tehnologiei curente și viitoare. Cerințele provocatoare ale diferitelor rețele cum ar fi Internetul sau transmisiunile fără fir necesită o privire proaspătă asupra strategiilor curente de codare și analizarea dacă ele sunt suficiente pentru aceste condiții. În fine, video este

aproape totdeauna însoțit de audio, iar nevoia de a studia compresia și transmisia combinată audio-video este o zonă de cercetare interesantă, care ar putea aduce multe valori tehnologiei viitoare.

Referințe bibliografice

1. A. J. Ahumada, Jr., "Putting the visual system noise back in the picture", *J. Optical Society of America A* (1987) 2372–2378.
2. J. K. Aggarwal și N. Nandhakumar, "On the computation of motion from sequences of images—A review", *Proc. IEEE* **76**, 8 (1988) 917–935.
3. N. Ahmed, T. Natarajan și K. Rao, "Discrete cosine transform", *IEEE Trans. Comput.* **23** (1974) 90–93.
4. K. Aizawa, H. Harashima și T. Saito, "Model-based analysis synthesis image coding for a person's face", *Image Communication* **1** (1989) 139–152.
5. K. Aizawa și T. S. Huang, "Model-based image coding: Advanced video coding techniques for very low bit-rate applications", *Proc. IEEE* **83** (1995) 259–271.
6. ANSI, "Digital processing of video signals—Video coder/decoder for audiovisual services at 56 to 1536 kbit/s", American National Standard T1.p64-199x, American National Standards Institute, Inc., oct. 1990.
7. M. Antonini, M. Barlaud, P. Mathieu și I. Daubechies, "Image coding using wavelet transform", *IEEE Trans. Image Proc.* (1992) 205–220.
8. J. Apostolopoulos și N. Jayant, "Post-processing for very-low-bit-rate video compression", înaintat către *IEEE Trans. on Image Processing.*
9. A. Aravind și A. Gersho, "Image compression based on vector quantization with finite memory", *Opt. Eng.* **26** (1987) 570–580.
10. A. Aravind, G. L. Cash, D. L. Duttweiler, H.-M. Hang, B. G. Haskell și A. Puri, "Image and video coding standards", *AT&T Technical J.* **72**, 1 (1993).
11. R. L. Baker și R. M. Gray, "Image compression using non-adaptive spatial vector quantization", în *Conf. Record of the Sixteenth Asilomar Conf. on Circuit Systems and Computers*, 1992.
12. R. L. Baker și R. M. Gray, "Differential vector quantization of achromatic imagery", în *Proc. Int. Picture Coding Symp.*, 1983.
13. F. Bosveld, R. L. Lagendijk și J. Biemond, "Hierarchical video coding using a spatio-temporal subband decomposition", în *Proc. ICASSP 92*, 1992.
14. Z. L. Budrikis, "Visual fidelity criterior and modeling", *Proc. IEEE* **60**, 7 (1972) 771–779.
15. P. J. Burt, "Smart sensing within a pyramid vision machine", *Proc. IEEE* (1988) 1006–1015.
16. P. J. Burt și E. H. Adelson, "The Laplacian pyramid as a compact image code", *IEEE Trans. Commun.* (1983) 532–540.
17. C. Cafforio și F. Rocca, "Methods for measuring small displacements of television images", *IEEE Trans. Information Theory* **IT-22** (1976) 573–579.
18. F. W. Campbell și J. G. Robson, "Application of Fourier analysis to the visibility of gratings", *J. Physiology* (1968) 551–566.
19. P. M. Cassereau, D. H. Staelin și G. de Jager, "Encoding of images based on a lapped orthogonal transform", *IEEE Trans. Commun.* **37** (1989) 189–193.
20. CCITT, "Video codec for audiovisual services at p*64 kbit/s", CCITT Recomendation H.261, CDM XV-R 37-E, International Telephone and Telegraph Consultative Committee, aug. 1990.
21. D. Chen și A. C. Bovik, "Visual pattern image coding", *IEEE Trans. Commun.* **38** (1990) 2137–2145.

22. D. Chen şi A. C. Bovik, "Hierarchical visual pattern image coding", *IEEE Trans. Commun.* **40** (1992) 671–675.
23. H. H. Chen, M. Reha Civanlar şi B. G. Haskell, "A block transform coder for arbitrarily-shaped image segments", în *Proc. of the Int. Workshop on Coding Techniques for Very Low Bit-rate Video*, 1994.
24. T. Chen, H. P. Graf şi K. Wang, "Speech-assisted video processing: Interpolation and low-bitrate coding", în *Proc 28th Asilomar Conf.*, 1994.
25. T. Chen şi P. P. Vaidyanathan, "Multidimensional multirate filters and filter banks derived from one-dimensional filters", *IEEE Trans. Signal Proc.*, 1993, 1749–1765.
26. W.-H. Chen şi W. K. Pratt, "Scene adaptive coder", *IEEE Trans. on Commun.* (1984) 225–232.
27. B. Chitprasert şi K. R. Rao, "Human visual weighted progressive image transmission", *IEEE Trans. on Commun.* (1990) 1040–1044.
28. P. L. Chu, "Quadrature mirror filter design for an arbitrary number of equal bandwidth channels", *IEEE Trans. ASSP* (1985) 203–218.
29. R. J. Clarke, "Relation between the Karhunen–Loeve and cosine transforms", *Proc. IEEE* **128** (1981) 359–360.
30. R. V. Cox, "The design of uniformly and nonuniformly spaced pseudoquadrature mirror filters", *IEEE Trans. ASSP* (1986) 1090–1096.
31. I. Craw, H. Ellis şi J. Lishman, "Automatic extraction of face features", *Pattern Recognition Lett.* **5** (1987) 183–187.
32. R. E. Crochiere, S. A. Webber şi J. L. Flanagan, "Digital coding of speech in subbands", *Bell System Technical J.* (1976) 1069–1085.
33. A. Croisier, D. Esteban şi C. Galand, "Perfect channel splitting by use of interpolation, decimation, tree decomposition techniques", în *Proc. Int. Conf. Inform. Sci./Syst.*, 1976.
34. V. Cuperman şi A. Gersho, "Adaptive differential vector coding of speech", în *Proc. Conf. Rec. GLOBECOM '82*, 1982, pp. 1092–1096.
35. S. Daly, "The visual difference predictor: An algorithm for the assessment of image fidelity", *Proc. of SPIE Conf. on Human Vision, Visual Processing and Digital Display III*, 1992, pp. 2–15.
36. I. Daubechies, "Orthonormal bases of compactly supported wavelets", *Commun. on Pure and Appl. Math.* **41** (1988) 909–996.
37. I. Daubechies, "The wavelet transform, time-frequency localization and signal analysis", *IEEE Trans. Information Theory.* **36** (1990) 961–1005.
38. E. J. Delp şi O. R. Mitchell, "Image compression using block truncation coding", *IEEE Trans. Commun.* (1979) 1335–1341.
39. N. Diehl, "Object-oriented motion estimation and segmentation in image sequences", *Signal Processing: Image Communication* **3** (1991) 23–56.
40. A. Eleftheriadis şi A. Jacquin, "Low bit rate model-assisted H.261-compatible coding of video", în *Proc. ICIP95*, 1995.
41. D. Esteban şi C. Galand, "Application of quadrature mirror filters to split band voice coding schemes", în *Proc. ICASSP*, 1978, pp. 191–195.
42. F. J. Kolb, Jr., redactor, "Bibliography: Psychophysics of image evaluation", *SMPTE J.* (1989) 594–599.
43. J. Foster, R. M. Gray şi M. O. Dunham, "Finite-state vector quantization for waveform coding", *IEEE Trans. Inform. Theory* (1985) 348–359.
44. T. Fukuhara, K. Asai şi T. Murakami, "Hierarchical division of 3-d wire-frame model and vector quantization in a model-based coding of facial images", în *Picture Coding Symp. '90*, 1990.

45. A. Gersho şi B. Ramamurthi, "Image coding using vector quantization", în *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1982, pp. 428–431.
46. A. Gersho şi R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.
47. H. Gharavi şi A Tabatabai, "Sub-band coding of monochrome and color images", *IEEE Trans. Circuits and Systems* (1988) 207–214.
48. B. Girod, "Psychovisual aspects of image communication", *Signal Processing* **28** (1992) 239–251.
49. B. Girod, "Motion-compensating prediction with fractional-pel accuracy", *IEEE Trans. Commun.* **41** (1993) 604–611.
50. W. M. Goodall, "Television by pulse code modulation", *Bell Systems Tech. J.* **30** (1951) 33–49.
51. R. E. Graham, "Predictive quantizing of television signals", *IRE Wescon Convention Record* **2** (1958) 147–157.
52. A. Haar, "Zur Theorie der orthogonalen Funktionen-System", *Math. Annalen* **5** (1955) 17–31.
53. A. Habibi şi G. S. Robinson, "A survey of digital picture coding", *Comput.* **7** (1974) 22–34.
54. H.-M. Hang, R. Leonardi, B. G. Haskell, R. L. Schmidt, H. Bheda şi J. Othmer, "Digital (HDTV) compression at 44 Mbps using parallel motion-compensated transform coders", în *Proc. of the SPIE Conf. on Visual Communications and Image Processing*, 1990.
55. H. M. Hang şi J. W. Woods, "Predictive vector quantization of images", *IEEE Trans. Commun.* **COM-33** (1985) 1208–1219.
56. B. G. Haskell, F. W. Mounts şi J. C. Candy, "Interframe coding of video telephone pictures", *Proc. IEEE* (1972) 792–800.
57. P Haskell şi D. Messerschmitt, "Resynchronization of motion compensated video affected by ATM cell loss", în *Proc. ICASSP*, 1992, pp. 545–548.
58. D. J. Healy şi D. R. Mitchell, "Digital video bandwidth compression using block truncation coding", *IEEE Trans. Commun.* **COM-29** (1981) 1809–1817.
59. H. Hotelling, "Analysis of a complex of statistical variables into principle components", *J. Educational Psychology* (1933) 417–441.
60. A. E. Jacquin, "A novel fractal block-coding technique for digital images", *Proc. ICASSP*, 1990, pp. 1625–2228.
61. A. K. Jain, "Image data compression: A review", *Proc. IEEE* **69**, 3 (1981) 349–391.
62. N. S. Jayant, J. D. Johnston şi R. J. Safranek, "Signal compression based on models of human perception", *Proc. IEEE*, oct. 1993.
63. N. S. Jayant şi P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice Hall, Inc., 1984.
64. J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks", în *Proc. ICASSP*, 1980.
65. H. Karhunen, "Über lineare Methoden in der Wahrscheinlichkeitsrechnung", *Ann. Acad. Sci. Fenn.* **Ser. A.I.37** (1947).
66. G. Karlsson şi M. Vetterli, "Three-dimensional subband coding of video", în *Proc. ICASSP*, 1988.
67. G. Karlsson şi M. Vetterli, "Theory of two-dimensional multirate filter banks", *IEEE Trans. Acoust., Speech, and Signal Proc.* (1990) 925–937.
68. M. D. Kelly, "Visual identification of people by computer", Technical Report A1-130, Stanford A1 Project, 1970.
69. T. Kim, "New finite state vector quantizers for images", în *Proc. ICASSP*, 1988.

70. T. Kim, "Side match and overlap match vector quantizers for images", *IEEE Trans. Image Processing* **1** (1992) 170–185.
71. Y.-H. Kim şi J. W. Modestino, "Adaptive entropy-coded subband coding of images", *IEEE Trans. Image Processing* (1992) 31–48.
72. S. A. Klein, A. D. Silverstein şi T. Carney, "Revelance of human vision to JPEG-DCT compression", *Proc. of SPIE Conf. on Human Vision, Visual Processing and Digital Display III*, 1992, pp. 200–215.
73. R. Koch, "Dynamic 3-D scene analysis through syntesis feedback control", *IEEE Trans. Patt. Anal. Mach. Intell.* **15** (1993) 556–568.
74. M. M. Kocher şi M. Kunt, "A contour-texture approach to picture coding", în *Proc. ICASSP '82*, 1992, pp. 436–440.
75. S. Komiyama, "Subjective evaluation of angular displacement between picture and sound directions for HDTV sound systems", *J. Audio Eng. Soc.* **37**, 4 (1989) 210–214.
76. J. Kovacevic, "Subband coding systems incorporating quantizer models", *IEEE Trans. Image Processing* (1995) 543–553.
77. J. Kovacevic şi M. Vetterli, "Non-separable multidimensional perfect reconstruction filter banks and wavelet bases for R^N", *IEEE Trans. Infor. Theory* **38** (1992) 533–555.
78. M. Kunt, A. Ikonomopoulos şi M. Kocher, "Second generation image-coding techniques", *Proc. IEEE* **73** (1985) 549–574.
79. W.-M. Lam, A. R. Reibman şi B. Liu, "Recovery of lost or errorneously received motion vectors", în *Proc. ICASSP*, 1993.
80. W.-M. Lam şi A. R. Reibman "An error concealment algorithm for images subject to channel errors", *IEEE Trans. on Image Processing* **4** (1995) 533–542.
81. D. LeGall, "MPEG, a video compression standard for multimedia applications", *Commun. ACM* **34** (1991) 47–58.
82. D. LeGall şi A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques", în *Proc. ICASSP '88*, 1988, pp. 761–764.
83. A. Lewis şi G. Knowles, "Video compression using 3D wavelet transforms", *Electronics Lett.* **26** (1990) 396–398; vezi şi pp. 1184–1185.
84. H. Li, A. Lundmark şi R. Forchheimer, "Image sequence coding at very low bitrates: A review", *IEEE Trans. Image Processing* **3**, 5 (1994) 589–609.
85. A. Ligtenberg şi M. Vetterli, "A discrete Fourier-cosine transform chip", *IEEE J. on Selected Areas in Commun.* **4** (1986) 49–61.
86. J. O. Limb, "Distortion criteria of the human viewer", *IEEE Trans. on Systems, Man, Cybernetics* (1979).
87. Y. Linde, A. Buzo şi R. M. Gray, "An algorithm for vector quantizer design", *IEEE Trans. Commun.* **COM-28** (1980) 84–95.
88. M. Liou, "Overview of the p*64 kbits/s video coding standard", *Commun. ACM* (1991) 60–63.
89. T. Liu şi N. Jayant, "Adaptive post-processing algorithms for low bit rate video signals", în *Proc. IEEE Conf. on Acoustics, Speech and Signal Processing*, aprilie 1994, pp. V-401–V-404.
90. S. P. Lloyd, "Least squares quantization in PCM", Institute of Mathematical Statistics Meeting, Atlantic City, NJ, 1957.
91. S. P. Lloyd, "Least squares quantization in PCM", *IEEE Trans. Inform. Theory* **IT-28** (1982) 127–135.
92. H. Lohscheller, "A subjectively adapted image communications system", *IEEE Trans. Commun.* (1984) 1316–1322.

93. J. Luo, C. Chen, K. Parker şi T. Huang, "A new method for block effect removal in low bitrate image compression", în *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, mai 1994, pp. V-341–V-344.
94. J. Luo, C. W. Chen, K. J. Parker şi T. S. Huang, "Three-dimensional subband video analysis and synthesis with adaptive clustering in high frequency subbands", în *IEEE Int. Conf. on Image Processing*, 1994.
95. J. MacQueen, "Some methods for classification and analysis of multivariate observations", în *Proc. of the Fifth Berkeley Symp. on Math. Stat. and Prob.*, vol. 1, 1967, pp. 281–296.
96. J. Makhoul, S. Roucos şi H. Gish, "Vector quantization in speech coding", *Proc. IEEE* (1985) 1551–1587.
97. S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of l^2 (r)", *Trans. Amer. Math. Soc.* (1989) 69–87.
98. S. Mallat, "A theory of multiresolution signal decomposition: the wavelet representation", *IEEE Trans. Patt. Recog. and Mach. Intell*; (1989) 674–693.
99. H. S. Malvar, *Signal Processing with Lapped Transforms,* Artech House, Inc., Norwood, MA, 1992.
100. H. S. Malvar şi D. H. Staelin, "The LOT: transform coding without blocking effects", *IEEE Trans. Acoust., Speech, Signal Processing* **37** (1989) 553–559.
101. J. L. Mannos şi D. J. Sakrison, "The effects of a visual fidelity criterion on the encoding of images", *IEEE Trans. on Information Theory* **IT-20**, 4 (1974).
102. D. Marr şi H. Hildreth, "Theory of edge detection", *Roy. Soc. London*, 1980, pp. 187–217.
103. J. Max, "Quantizing for minimum distortion", *IRE Trans. on Information Theory* (1960) 7-12.
104. D. L. McLaren şi D. T. Nguyen, "Removal of subjective redundancy from DCT-coded images", *Proc. IEEE: Part I* (1991) 345–350.
105. T. Minami, I. So şi O. Nakamura, "Knowledge-based coding of facial images", în *Picture Coding Symp. '90,* 1990.
106. O. R. Mitchell şi A. J. Tabatabai, "Channel error recovery for transform image coding", *IEEE Trans. Commun.* **29** (1981) 1754–1762.
107. J. W. Modestino şi D. G. Daut, "Combined source-channel coding of images", *IEEE Trans. Commun.* **COM-27** (1979) 1644–1659.
108. A. Moffat, T. C. Bell şi I. H. Witten, "Lossless compression for text and images", acest volum.
109. H. Morikawa şi H. Harashima, "3-D structure extraction coding of image sequences", *J. Visual Commun. and Image Representation* **2** (1991) 332–344.
110. F. W. Mounts, "A video encoding system with conditional picture-element replenishment", *Bell System Tech. J.* (1969) 2545–2554.
111. T. Murakami, K. Asai şi E. Yamazaki, "Vector quantizer of video signals", *Electron. Lett.* **7** (1982) 1005–1006.
112. H. G. Musmann, *Predictive Image Coding,* Academic Press, New York, 1979.
113. H. G. Musmann, M. Hotter şi J. Ostermann, "Object-oriented analysis-synthesis coding of moving images", *Image Commun.* **1** (1989) 117–138.
114. H. G. Musmann, P. Pirsch şi H.-J. Grallert, "Advances in picture coding", *Proc. IEEE* **73**, 4 (1985) 523–548.
115. Y. Nakaya, K. Aizawa şi H. Harashima, "Texture updating methods in modelbased coding of facial images", în *Picture Coding Symp. '90,* 1990.

116. Y. Nakaya şi H. Harashima, "Model-based/waveform hybrid coding for low-rate transmission of facial images", *IEICE Trans. Commun.* **E75-B** (1992) 377–384.
117. Y. Nakaya şi H. Harashima, "Motion compensation based on spatial transformations", *IEEE Trans. Video Technol.* 4 (1994) 339–356.
118. N. M. Nasrabadi şi R. A. King, "Image coding using vector quantization: A review", *IEEE Trans. on Commun.* **36** (1988) 957–971.
119. A. Netravali şi B. Haskell, *Digital Pictures, Representation and Compression,* Plenum, New York, 1988.
120. A. N. Netravali şi L. O. Limb, "Picture coding: A review", *Proc. IEEE* **68** (1980) 366-406.
121. A. N. Netravali, E. Petajan, S. Knauer, K. Mathews, R. J. Safranek şi P. Westerink, "A high quality digital HDTV codec", *IEEE Trans. Consumer Electronics* (1991) 320-330.
122. A. N. Netravali şi J. B. Robbins, "Motion-compensated television coding: Part I", *Bell Syst. Tech. J.* **58** (1979) 631-670.
123. A. N. Netravali şi J. D. Robbins, "Motion-compensated coding: Some new results", *Bell Syst. Tech. J.* **59** (1980) 1735-1745.
124. D. L. Neuhoff şi T. N. Pappas, "Perceptual coding of images for halftone display", *IEEE Trans. Image Proc.* **IP-3** (1994).
125. D. L. Neuhoff şi N. Moayeri, "Tree searched vector quantization with interblock noiseless coding", în *Proc. 1988 IEEE ICISS,* 1988.
126. K. N. Ngan, K. S. Leong şi H. Singh, "Adaptive cosine transform coding of images in perceptual domain", *IEEE Trans. ASSP* (1989) 1743-1750.
127. King N. Ngan şi Weng L. Chooi, "Very low bit rate video coding using 3D subband approach", *IEEE Trans. Circuits and Systems for Video Technology* **4** (1994).
128. N. B. Nill, "A visual model weighted cosine transform for image compression and quality assessment", *IEEE Trans. on Commun.* **551** (1985).
129. Y. Ninomiya şi Y. Ohtsuka, "A motion-compensated interframe coding scheme for television pictures", *IEEE Trans. Commun.* **COM-30** (1982) 201-211.
130. B. Noble şi J. W. Daniel, *Applied Linear Algebra,* Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977.
131. J.-R. Ohm, "Three-dimensional subband coding with motion compensation", *IEEE Trans. Image Processing* **3**, 5 (1994).
132. M. T. Orchard, "Predictive motion-field segmentation for image sequence coding", *IEEE Trans. Circuits and Systems for Video Technology* **3** (1993) 54-70.
133. J. Ostermann, "Modelling of 3-D moving objects for an analysis-synthesis coder", în *Proc. SPIE, Sensing and Reconstruction of Three-Dimensional Objects and Scenes,* vol. 1260, 1990.
134. T. N. Pappas, *Perceptual Coding and Printing of Gray-Scale and Color Images,* SID-92, Digest of Technical Papers, 1992.
135. T. N. Pappas şi D. L. Neuhoff, "Printer models and error diffusion", *IEEE Trans. Image Proc.* **4** (1995).
136. W. B. Pennebaker şi J. L. Mitchell, *JPEG Still Image Data Compression Standard,* Van Norstrand Rienhold, New York, 1993.
137. M. G. Perkins şi T. Lookabaugh, "A psychophysically justified bit allocation algorithm for subband image coding systems", *Proc. ICASSP,* 1989.
138. H. A. Peterson, A. J. Ahumada, Jr. şi A. B. Watson, "Improved detection model for DCT coefficient quantization", *SPIE Conf. on Human Vision, Visual Processing, and Digital Display IV* **1913** (1993) 191-201.

139. C. Podilchuk şi A. Jacquin, "Geometric vector quantization for subband-based video coding", în *Proc. of DIMACS/IEEE Workshop on Coding and Quantization,* 1992.
140. C. Podilchuk, "Low-bit rate subband video coding", în *Proc. ICIP 94,* 1994.
141. C. Podilchuk şi A. Jacquin, "Subband video coding with a dynamic bit allocation and geometric vector quantization", în *Proc. SPIE-Human Vision, Visual Processing and Digital Display III,* 1992.
142. C. I. Podilchuk, N. S. Jayant şi N. Farvardin, "Three-dimensional subband coding of video", *IEEE Trans. on Image Processing* **4** (1993) 125-139.
143. C. I. Podilchuk şi N. Farvardin, "Perceptually based low bit rate video coding", în *Proc. ICASSP,* 1991.
144. C. I. Podilchuk, N. S. Jayant şi P. Noll, "Sparse codebooks for the quantization of non-dominant subbands in image coding", în *Proc. ICASSP,* 1990.
145. W. K. Pratt, H. C. Andrews şi J. Kane, "Hadamard transform image coding", *Proc. IEEE* **57** (1969) 58–68.
146. W. K. Pratt, W. H. Chen şi L. R. Welch, "Slant transform image coding", *IEEE Trans. Commun.* **COM-22** (1974) 1075–1093.
147. J. Princen, A. Johnson şi A. Bradley, "Sub-band transform coding using filterbank designs based on time-domain aliasing cancellation", *Proc. ICASSP,* 1987, pp. 2161–2164.
148. A. Puri şi R. Aravind, "Motion-compensated video coding with adaptive perceptual quantization", *IEEE Trans. Circuits and Systems for Video Technology* **1**, 4 (1991).
149. B. Ramamurthi şi A. Gersho, "Classified vector quantization of images", *IEEE Trans. Commun.* **COM-34** (1986) 1105–1115.
150. B. Ramamurthi şi A. Gersho, "Nonlinear space-variant postprocessing of block coded images", *IEEE Trans. on Acoustics, Speech and Signal Processing* **ASSP-34** (1986) 1258–1268.
151. K. R. Rao şi P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications,* Academic Press, 1990.
152. H. Reeve şi J. Lim, "Reduction of blocking effects in image coding", *Optical Engineering* **23** (1984) 34–37.
153. O. Rioul şi M. Vetterli, "Wavelets and signal processing", *IEEE Signal Processing Mag.* **8** (1991) 14–38.
154. L. G. Roberts, "Picture coding using pseudo-random noise", *IRE Trans. Information Theory* (1962) 145–154.
155. J. H. Rothweiler, "Polyphase quadrature filters–a new subband coding technique", în *Proc. ICASSP,* 1983, pp. 1280–1283.
156. R. J. Safranek, "Perceptually based prequantization for image compression", *Proc. of SPIE Conf. on Human Vision, Visual Processing and Digital Display V,* 1994.
157. R. J. Safranek, "A comparison of the coding efficiency of perceptual models", *Proc. of SPIE Conf. on Human Vision, Visual Processing and Digital Display VI,* 1995.
158. R. J. Safranek şi J. D. Johnston, "A perceptually tuned subband image coder with image-dependent quantization and post-quantization data compression", *Proc. ICASSP,* 1989.
159. R. J. Safranek, J. D. Johnston şi R. E. Rosenholtz, "A perceptually tuned subband image coder", prezentat la *SPIE Symp. on Electronic Imaging Human Vision & Electronic Imaging: Models Methods and Applications,* Santa Clara, CA, feb. 1990.
160. R. J. Safranek, J. D. Johnston, N. S. Jayant şi C. I. Podilchuk, "Perceptual coding of image signals", în *Asilomar Conf. Proc.,* 1990.

161. J. A. Saghri, P. S. Cheatham şi A. Habibi, "Image quality measure based on a human visual system", *Optical Engineering,* iulie 1989, pp. 813–818.
162. T. Saito, H. Takeo, K. Aizawa, H. Harashima şi H. Miyakawa, "Adaptive discrete cosine transform image coding using gain/shape vector quantization", în *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing,* 1986, pp. 129–132.
163. T. Sakai, M. Nagao şi S. Fujibayashi, "Line extraction and pattern recognition in a photograph", *Pattern Recognition* **1** (1969) 233–248.
164. W. F. Schreiber, "Picture coding", *Proc. IEEE* **55** (1967) 320–330.
165. W. F. Schreiber, T. S. Huang şi O. J. Tretiak, "Contour coding of images", *Picture Bandwidth Compression,* Gordon and Breach, 1972.
166. J. M. Shapiro, "Application of the embedded wavelet hierarchical image coder to very low bit rate image coding", în *Proc. ICASSP '93,* 1993, pp. 558–561.
167. J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coeficients", *IEEE Trans. Signal Processing* **41** (1993) 3445–3462.
168. E. P. Simoncelli şi E. H. Adelson, "Nonseparable extensions of quadrature mirror filters to multiple dimensions", *Proc. IEEE,* vol. 78, 1990, pp. 652–664.
169. M. J. T. Smith şi S. L. Eddins, "Subband coding of images with octave band tree structures", în *Proc. ICASSP '87,* 1987.
170. M. J. T. Smith şi T. P. Barnwell III, "A new filter-bank theory for time-frequency representation", *IEEE Trans. ASSP* **35** (1987) 314–327.
171. R. Srinivasan şi K. R. Rao, "Predictive coding based on efficient motion estimation", *IEEE Trans. Commun.* **COM-33** (1985) 888–896.
172. C. F. Stromeyer şi B. Julesz, "Spatial frequency masking in vision: Critical bands and spread of masking", *J. Opt. Soc. America* (1972) 1221–1232.
173. N. Tanabe şi N. Farvardin, "Subband image coding using entropy-coded quantization over noisy channels", *IEEE J. Select. Areas Commun.* **10** (1992) 926–943.
174. D. Taubman şi A. Zakhor, "Multirate 3-D subband coding of video", *IEEE Trans. Image Processing* **3,** 5 (1994) 572–588.
175. P. P. Vaidyanathan, "Quadrature mirror filter banks, *m*-band extensions and perfect reconstruction techniques", *IEEE ASSP Mag.* **4** (1987) 4–20.
176. P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: A tutorial", *Proc. IEEE* **78** (1990) 56–93.
177. P. P. Vaidyanathan, *Multirate Systems and Filter Banks,* Prentice Hall, 1993.
178. V. A. Vaishampayan şi N. Farvardin, "Optimal block cosine transform image coding for noisy channels", *IEEE Trans. Commun.* **38** (1990) 327–336.
179. R. E. Van Dyck şi S. A. Rajala, "Subband/VQ coding in perceptually uniform color spaces", *Proc. ICASSP,* 1992, pp. III-237–III-240.
180. M. Vetterli, "Multidimensional subband coding: Some theory and algorithms", *Signal Processing* **6** (1984) 97–112.
181. M. Vetterli şi H. J. Nussbaumer, "Simple FFT and DCT algorithms with reduced number of operations", *Signal Processing* **6** (1984) 267–278.
182. M. Vetterli şi J. Kovacevic, *Wavelets and Subband Coding,* Prentice Hall, 1995.
183. G. K. Wallace, "The JPEG Still Picture Compression Standard", *Commun. A CM* **34** (1991) 31–43.
184. Y. Wang şi V. Ramamoorthy, "Image reconstruction from partial subband images and its applications in packet video transmission", *Signal Proc.: Image Commun.* **3** (1991) 197–230.
185. Z. Wang, "Fast algorithms for the discrete *w* transform and for the discrete Fourier transform", *IEEE Trans. Acoust., Speech, Signal Processing* **32** (1984) 803–816.

186. A. B. Watson, "Efficiency of an image code based on human vision", *J. Optical Society of America A* (1987) 2401–2417.
187. A. B. Watson, "The Cortex transform: Rapid computation of simulated neural images", *Computer Vision, Graphics, and Image Processing,* 1987, pp. 311–327.
188. A. B. Watson, "DCT quantization matricies visually optimized for individual images", *Proc. of SPIE Conf. on Human Vision, Visual Processing and Digital Display IV,* 1992, pp. 202–216.
189. P. H. Westerink, J. Biemond şi D. E. Boekee, "Scalar quantization error analysis for image subband coding using QMF's", *IEEE Trans. Signal Processing* **40** (1992) 421–428.
190. P. H. Westerink, J. Biemond, D. E. Boekee şi J. W. Woods. "Subband coding of images using vector quantization", *IEEE Trans. Commun.* **36** (1988) 713–719.
191. P. H. Westerink, J. Biemond şi F. Muller, "Subband coding of image sequences at low bit rates", *Signal Processing: Image Communication* (1990) 441–448.
192. J. W. Woods, *Subband Image Coding,* Kluwer Academic Publishers, 1991.
193. J. W. Woods şi S. D. O'Neil, "Sub-band coding of images", *IEEE Trans. ASSP* **34** (1986) 1278–1288.
194. Z. Xie şi T. G. Stockham, Jr., "Previsualized image vector quantization with optimized pre- and postprocessors", *IEEE Trans. on Commun.* **39** (1991) 1662–1671.
195. Y. Yamada, K. Fujita şi S. Tazaki, "Vector quantization of video signals", în *Proc. Annual Conf. of IECE,* 1980, p. 1031.
196. Y. Yang, N. Galatsanos şi A. Katsaggelos, "Iterative projection algorithms for removing the blocking artifacts of block-dct compressed images", în *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing,* aprilie 1993, pp. V-405–V-408.
197. Y. Yang, N. Galatsanos şi A. Katsaggelos, "Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images", *IEEE Trans. on Circuits and Systems for Video Technology* **3** (1993) 421–432.
198. A. Yuille, D. Cohen şi P. Hallinan, "Feature extraction from faces using deformable templates", în *Proc. IEEE CSC on Computer Vision and Pattern Recognition,* 1989, pp. 104–109.

International Journal of High Speed Electronics and Systems, Vol.8, No. 1 (1997) 179-231
© World Scientific Publishing Company

COMPRESIA FĂRĂ PIERDERI PENTRU TEXT ȘI IMAGINI

ALISTAIR MOFFAT
Department of Computer Science,
The University of Melbourne, Parkville, Australia 3052
E-mail: alistair@cs.mu.oz.au

TIMOTHY C. BELL
Department of Computer Science,
University of Canterbury, New Zealand
E-mail: tim@cosc.canterbury.ac.nz

IAN H. WITTEN
Department of Computer Science,
University of Waikato, New Zealand
E-mail: ihw@waikato.ac.nz

Majoritatea datelor, în mod inerent discrete, au nevoie de compresie efectuată într-un astfel de mod încât să poată fi reconstituite exact, fără nici o pierdere. Exemplele includ toate tipurile de texte, rezultate experimentale și baze de date statistice. Alte forme de date au nevoie să fie înmagazinate exact, ca de exemplu imaginile, în mod particular cele cu două niveluri, sau unele generate în medicină sau aplicații de percepție la distanță, sau unele cărora li se cere certificarea pentru rațiuni de legalitate. În plus, în timpul procesului de compresie curentă cu pierdere de informație, se ivesc multe ocazii pentru compresia fără pierderi a coeficienților sau a altor informații.

Această lucrare examinează tehnicile de compresie fără pierderi. Procesul de compresie poate fi descompus în modelare și codare. Furnizăm o discuție extensivă asupra tehnicilor de codare și prezentăm apoi metode de modelare adecvate pentru text și imagini. Sunt descrise metode standard utilizate în utilități obișnuite (pentru cazul textului) și standarde internaționale (în cazul imaginilor).

Conținut

1. Introducere 173
2. Modelare și codare 174
   2.1. Separarea funcțiilor 174
   2.2. Trecere în revistă a metodelor de codare 176
   2.3. Codarea non-entropică 176
   2.4. Codarea entropică 177
   2.5. Codarea de minimă redundanță 178
   2.6. Codarea aritmetică 183

2.7. Performanțe relative 187
2.8. Codarea alfabetelor binare 188
3. Modele pentru text 189
3.1. Modele contextuale 189
3.2. Modele dicționar 190
3.3. Echivalența modelelor 193
3.4. Algoritmul PPM 194
3.5. Algoritmul ZIP 196
3.6. Algoritmul COMPRESS 197
3.7. Performanța relativă 198
3.8. Viitorul compresiei de text 199
4. Modele pentru imagini 199
4.1. Standardul CCITT pentru facsimil 200
4.2. Modele contextuale 202
4.3. Modele cu două niveluri 204
4.4. JBIG: Un standard pentru imagini cu două niveluri 205
4.5. Formatul GIF pentru compresia fără pierderi a imaginii 208
4.6. Schema FELICS pentru compresia imaginilor cu scară de gri 209
4.7. Schema CALIC pentru compresia imaginilor de tonuri continue 211
4.8. Performanțele metodelor de compresie fără pierderi 212
5. Alte forme de date 214
5.1. Compresia sunetului 214
5.2. Compresia indexurilor 215
5.3. Imagini textuale 215
5.4. Compresia fișierelor în sistem 216
6. Concluzie 217
Referințe bibliografie 218

1. Introducere

Mare parte din datele stocate în sistemul unui calculator sunt reprezentarea digitală a unei surse care, prin însăși natura ei, este continuă. Secvențe video, imagini scanate, și date de sunet – sunt toate în această categorie. Întrucât forma stocată este deja o versiune cuantizată a originalului, ea este adecvată permiterii unor aproximări ulterioare și pot fi utilizate tehnici de compresie cu pierderi (*lossy*) pentru a se obține reprezentări extrem de compacte. Utilizatorul este astfel liber să aleagă o rată de compresie – o apăsare pe „butonul" compresorului – care oferă echilibrul dorit între fidelitatea datei reconstruite și costul stocării. Din acest motiv preocuparea cercetării în domeniul compresiei de date video, de imagine și sunet se îndreaptă spre metodele cu pierderi, scopul fiind un compromis optim.

Pe de altă parte, anumite tipuri de date sunt în mod intrinsec discrete sau digitale. Textul stocat reprezintă exemplul arhetip în acest sens, dar sunt multe alte situații în care trebuie să existe posibilitatea refacerii cu exactitate a sursei originale. Aceste aplicații includ date de imagine care trebuie certificate din rațiuni legale sau medicale sau date de tipul imaginilor percepute de la distanță pentru care utilizarea finală este încă necunoscută și pentru care refacerea cu aproximație poate fi inadecvată. În cazul

imaginilor transmise prin satelit este, de asemenea, o chestiune de cost – este de departe mai puțin costisitor să stochezi astfel de imagini decât să le creezi și compresia cu pierderi poate fi foarte bine o falsă economie. Arhivarea imaginilor document istoric este iarăși posibil să pretindă compresie fără pierderi – nevoile viitorilor cercetători în domeniu neputând fi anticipate. Pentru rațiuni similare, stocarea datelor statistice – de tipul tabulațiilor – trebuie să fie exactă. Și, ca un exemplu final, ne putem imagina haosul care ar putea domni dacă un program executabil supus compresiei ar fi decomprimat într-o formă ușor diferită față de original.

În această lucrare examinăm metodele curent utilizate pentru compresia fără pierderi a datelor. Ne focalizăm atenția asupra a trei tipuri principale de date, dar metodele pe care le descriem sunt cu aplicabilitate generală. Într-adevăr, una din caracteristicile marcante pentru metodele de compresie fără pierderi este aceea că permite compresia oricărui fișier – și apoi refacerea sa – folosind orice metodă de compresie. Singura variabilă este gradul compresiei; întrucât compresia este fără pierderi, reproducerea fidelă este întotdeauna asigurată.

Acest articol este structurat în patru capitole. În capitolul 2 descriem paradigma modelare – codare pentru compresia fără pierderi și descriem o serie de metode de codare. Capitolul 3 arată apoi aplicabilitatea acestor idei la date de text, incluzând o discuție asupra a două metode actuale de compresie. În partea a 4-a examinăm compresia fără pierderi a imaginilor. Aici sunt descrise câteva metode uzuale – inclusiv standardul JBIG pentru reprezentările imaginilor binare (cu două niveluri) și Grupul 4 de standarde pentru facsimil. O serie de metode de compresie pentru alte tipuri de date sunt prezentate pe scurt în capitolul 5. La sfârșit poate fi găsită o bibliografie cuprinzând cele mai relevante cercetări în domeniu.

2. Modelare și codare

Un aspect central în înțelegerea metodelor curente de compresie fără pierderi îl reprezintă paradigma „modelare, codare". În acest capitol descriem acest cadru de lucru și dăm detalii asupra metodelor de codare. Capitolele 3, 4 și 5 descriu modele adecvate diferitelor tipuri de date și sugerează care din aceste codoare pot fi cuplate acestora în cel mai folositor mod.

2.1. *Separarea funcțiilor*

Figura 1 reprezintă structura logică a unui sistem adaptiv de compresie fără pierderi. Partea stângă a schemei arată acțiunea codorului; partea centrală reprezintă un canal de comunicații fără zgomot; iar partea dreaptă descrie decodorul.

La fiecare pas de codare, *modelul* este necesar pentru a estima o probabilitate pentru fiecare simbol posibil, fără nici o cunoaștere a simbolului care urmează a fi reprezentat. Modulul de *codare* preia această probabilitate de distribuție, împreună cu cunoașterea următorului simbol din fluxul de date și transmite către decodor o secvență de biți (sau de simboluri, într-un alt alfabet de canal). Doar după ce codul pentru simbol a fost transmis este posibil ca etajul de actualizare al modelului – faza de *estimare probabilistică* sau *statistică* – să decidă ce schimbări să se aplice modelului. Exact posibilitatea de a aplica aceste schimbări face codoarele adaptive atât de perfor-

mante. Schimbările pot fi făcute în probabilitatea simbolurilor, pentru a încorpora statisticile actualizate care rezultă după ce a fost văzut acest singur simbol; şi ele pot fi făcute în însăşi structura modelului.

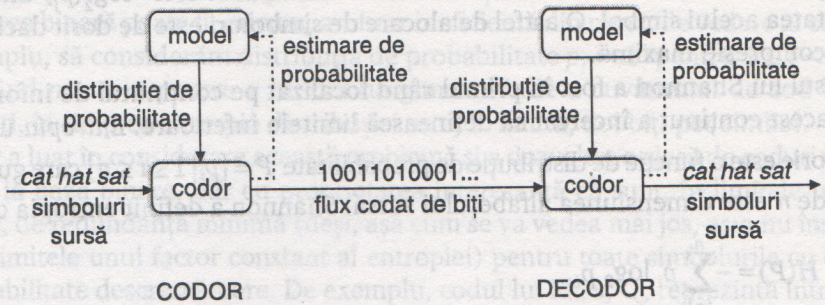


Figura 1. Structura unui compresor fără pierderi

În partea de decodare, modelul trebuie să furnizeze o probabilitate de distribuţie identică modulului de codare, care apoi extrage şi interpretează un număr adecvat de biţi din fluxul de biţi transmis şi determină simbolul corespondent din alfabetul sursă. Acest simbol este folosit în faza de estimare a probabilităţii pentru a reproduce orice schimbări efectuate în modelul de la codor, prin aceasta menţinându-se sincronizarea modelului. Întrucât codorul şi decodorul au acum modele identice, următorul simbol poate fi procesat în aceeaşi manieră. Funcţia modelului este, prin urmare, de a face cu acurateţe predicţii ale probabilităţilor simbolurilor care urmează în fluxul de date, în ipoteza cunoaşterii tuturor simbolurilor codate până atunci şi, poate a unei informaţii comune, a priori repartizată atât codorului cât şi decodorului.

Funcţia modelului în faza de estimare a probabilităţii este de a revizui estimările de probabilitate stocate – fie explicit, fie implicit – în model. Această revizuire poate fi la fel de simplă ca şi incrementarea unui numărător care înregistrează frecvenţa simbolului transmis; sau poate să implice ajustări considerabile ale structurii modelului şi crearea unei noi clase de condiţionare pe care să se bazeze estimările viitoare.

Funcţia modulului de codare este clară şi este cea mai „mecanică" dintre cele trei procese. Fiind dat un simbol particular şi o distribuţie de probabilitate pe întregul alfabet de simboluri, codorul trebuie să determine şi să transmită codul corespunzător. Fiind dată aceeaşi distribuţie de probabilitate şi fluxul de date codate, decodorul trebuie să determine care simbol a fost codat.

Această separare a funcţiilor – model şi codor, în care prima include estimarea explicită a probabilităţii – a fost pentru prima dată expusă de Rissanen şi Langdon.[1] Din motive de eficienţă a implementării, în sistemele practice de compresie, distincţia poate să nu fie la fel de bine făcută ca în descrierea de aici. Oricum, pentru aproape toate metodele de compresie cunoscute se poate determina o metodă echivalentă bazată pe simboluri, în care probabilitatea pentru fiecare simbol este determinată şi este folosit un codor pentru a reprezenta acel simbol. Langdon[2] dă un exemplu de astfel de echivalenţă; altele se pot găsi în lucrarea [3].

2.2. *Trecere în revistă a metodelor de codare*

Conform teoremei de codare sursă a lui Shannon,[4] codarea optimă se obține atunci când lungimea codului alocat celei de-a i-a litere din alfabet este $-\log_2(p_i)$ unde p_i este probabilitatea acelui simbol. O astfel de alocare de simboluri este de dorit dacă vrem să obținem compresie maximă.

Interesul lui Shannon a fost în primul rând focalizat pe conținutul de informație și, folosind acest conținut a încercat să definească limitele inferioare. *Entropia* unei surse fără memorie este o funcție de distribuție de probabilitate $P = [p_i | 1 \le i \le n]$ care guvernează sursa, unde n este dimensiunea alfabetului sursă. Shannon a definit entropia ca fiind

$$H(P) = -\sum_{i=1}^{n} p_i \log_2 p_i.$$

Măsurată în biți pe simbol, entropia definește o limită a compresibilității pentru acea sursă, care nu poate fi depășită de nici un codor. Vom descrie mai jos un codor care atinge esențialmente această limită.

Există, oricum, mai multe obiective posibile pentru proiectarea unui codor. O caracteristică de dorit este viteza de operare – ar trebui să se codeze și decodeze rapid. O alta este economia memoriei utilizate, astfel încât codarea și decodarea să poată fi făcute fără a folosi mari arii de memorie. Datorită tensiunilor dintre aceste cerințe conflictuale, nu există o singură metodă „cea mai bună" de codare. Există, oricum, un număr relativ mic de alternative concurente și acestea sunt revăzute în subcapitolele care urmează. O cale de a clasifica metodele alternative este împărțirea în codoare *entropice* și *non-entropice*. Primele urmăresc viteza de operare sau economia de memorie, ultimele urmăresc compactitatea reprezentării comprimate.

O altă cale importantă de caracterizare a codoarelor este după faptul că ele sunt sau nu *statice* sau *dinamice*, adică dacă suportă sau nu modele adaptive. Dacă se folosește un model adaptiv, distribuția de probabilitate este dependentă de schimbările continue, iar pre-calcularea cuvinetlor de cod este imposibilă. Diferite metode de codare reacționează diferit la cerințele modelelor adaptive, iar cunoașterea modelului este un factor important în alegerea codorului, dacă viteza de compresie trebuie maximizată, iar această distincție este, de asemenea, explorată în cele ce urmează.

În sfârșit, este convenabil să presupunem că P, lista probabilităților celor n simboluri, este furnizată într-o ordine monotonă, fie necrescătoare, fie nescăzătoare. Nu există nici o situație în care să se obțină vreun avantaj de compresie plecând de la o presupunere diferită, iar ceea ce se poate face mai bine în orice implementare este să așezăm simbolurile care sunt de fapt reprezentate – caractere, cuvinte și așa mai departe – într-o listă ordonată de identificatori.

2.3. *Codarea non-entropică*

Codoarele non-entropice folosesc aceeași reprezentare fără a ține seama de distribuția de probabilitate a simbolurilor care se codează. Cel mai simplu și mai rapid codor non-entropic folosește numere binare convenționale, luând, $\lceil \log_2 n \rceil$ biți per cuvânt de cod, pentru a coda fiecare simbol dintr-un alfabet de n simboluri. Acest codor necesită numai câteva cuvinte de memorie pentru a fi implementat. O îmbunătățire în eficiența

compresiei poate fi obținută dacă cuvintelor de cod pentru cele mai probabile $2^{\lceil \log_2 n \rceil} - n$ simboluri li se alocă lungimi de $\lfloor \log_2 n \rfloor$ biți, iar schimbarea are un impact modest asupra volumului de memorie sau asupra debitului de ieșire.

Codarea binară poate fi, oricum, arbitrar ineficientă din punct de vedere al compresiei. De exemplu, să considerăm distribuția de probabilitate $p_i = 2^{-i}$ pentru $1 \leq i < n$, și $p_n = 2^{-(n-1)}$. Când n este o putere a lui doi lungimea medie a cuvântului de cod binar este $\log_2 n$ biți, în timp ce entropia distribuției este, la limită, doi biți per simbol.

Elias[5] a luat în considerare această problemă și a dezvoltat o clasă de coduri universale inegale, la bază binare, dar cu proprietatea interesantă că sunt, în limitele unui factor constant, de redundanță minimă (deși, așa cum se va vedea mai jos, asta nu înseamnă că sunt în limitele unui factor constant al entropiei) pentru toate simbolurile cu distribuție de probabilitate descrescătoare. De exemplu, codul lui Elias, C_γ, reprezintă întregul $x \geq 1$ ca $\lfloor \log_2 x \rfloor$ în unar de origine zero, luând $1 + \lfloor \log_2 x \rfloor$ biți; urmat de $x - 2^{\lfloor \log_2 x \rfloor}$ în binar de origine zero, luând $\lfloor \log_2 n \rfloor$ biți. Asta înseamnă că, întregul x este codat folosind $1 + 2\lfloor \log_2 x \rfloor$ biți. O trăsătură utilă a acestui cod este că are terminația deschisă și nu există nici o cerință pentru codor sau decodor să cunoască pe n, numărul maxim al semnelor alfabetului. Acest cod (și celelalte coduri definite de Elias) poate astfel să fie util pentru alfabete infinite ca și pentru alfabete finite.

Golomb[6], Gallager și Van Voorhis[7] au considerat de asemenea distribuții infinite. Ei au descris un cod parametrizat care din nou permite reprezentarea valorilor arbitrar de mari. Fiind dat parametrul b, întregul $x > 0$ este codat ca $(x\text{-}1)$ div b într-un cod unar de origine zero, urmat de $(x\text{-}1)$ mod b într-un cod binar de origine zero (folosind aceeași modificare în binar menționată anterior) pentru întregii $0 \ldots b - 1$. De exemplu, cu $b = 6$ codul pentru $x = 10$ este „10101", adică, „10" reprezentând 1 în unar de origine zero și „101" ca un cod binar ajustat pentru 3. Cazul special când b este o putere a lui doi este cunoscut uneori sub numele de cod Rice.[8]

Codurile Golomb au o proprietate destul de surprinzătoare. Să presupunem că un eveniment favorabil apare cu o probabilitate p – de exemplu, când dăm cu banul și „capul" este fața favorabilă. Să presupunem că fețele sunt independente. Atunci, probabilitatea de a avea exact $x - 1$ evenimente nefavorabile înainte de următorul eveniment favorabil este dată de distribuția geometrică, $Pr(x) = (1-p)^{x-1}p$. Rezultă că dacă parametrul Golomb b este ales drept $\lceil -\log(2-p)/\log(1-p) \rceil$ atunci codul Golomb rezultant este un cod de minimă redundanță (definit mai jos) pentru probabilitățile distribuției geometrice.

Ca și codurile binare, codurile Golomb și Rice pot fi generate extrem de rapid. Într-adevăr, deoarece ele au în mare același cost de calcul per bit de ieșire ca și codurile binare, pot de fapt să ofere viteze mai mari decât binarele, deoarece pentru multe distribuții de probabilitate conduc la mai puțini biți la ieșire. Această combinație de atribute face codurile Golomb extrem de atractive pentru aplicații în care vectori de biți de densitate cunoscută (și mică) trebuie reprezentați și se poate asuma independența lor. Aceste aplicații includ compresia imaginilor, discutată în capitolul 4 și indexurile bazelor de date, discutate în capitolul 5.

2.4. *Codarea entropică*

Dacă scopul este de a maximiza compresia – adică de a minimiza spațiul cerut de forma codată – ar trebui folosit un codor entropic. Pentru claritate, vom împărți codoarele entropice în două familii. Întâi vom examina familia codoarelor de *redundanță minimă*,

al cărei exemplu clasic este algoritmul Huffman.[a] Dacă trebuie alocate cuvinte de cod de lungime integrală simbolurilor individuale ale alfabetului, aceste coduri oferă cea mai bună compresie posibilă, de aici și denumirea de „minimă redundanță".

În ciuda caracterizării de „cele mai bune", codoarele cu minimă redundanță pot fi încă arbitrar ineficiente. Să considerăm un alfabet de două simboluri, cu $p_1=2^{-k}$ și $p_2=1-2^{-k}$. Evident, orice cod de lungime integrală pentru acest alfabet trebuie să aloce „0" pentru un simbol și „1" pentru celălalt. Și astfel, costul mediu este de un bit per simbol. Entropia alfabetului poate, oricum, să fie adusă mai aproape de zero în măsura impusă de o alegere corespunzătoare a lui k.

O compresie aproape optimală este posibilă dacă cerința de lungime integrală a cuvintelor de cod este abandonată și un exemplu al acestei familii de codoare *aritmetice* este de asemenea descris mai jos. Deși pare surprinzător, un codor aritmetic poate de fapt să aloce cuvinte de cod de lungimi aproape exact egale cu $\log_2 p_i$, folosind fracțiuni de bit.

2.5. *Codarea de minimă redundanță*

Huffman[9] a folosit o paradigmă particulară pentru a proiecta un algoritm care construiește coduri de minimă redundanță. Înainte de a descrie acest algoritm, este util să ne îndreptăm puțin atenția asupra a ceea ce trebuie de fapt calculat. În dezvoltarea anterioară am privit setul de *cuvinte de cod* ca fiind „codurile" și am presupus că procesul de construcție a codului era procesul de alocare a unui cuvânt de cod binar fiecărui simbol.

În general, pentru ca acest cod să fie unic decodabil, codul trebuie să fie *fără prefixe*. Adică nici un cuvânt de cod nu poate fi prefixul unui alt cuvânt de cod. (Astfel de coduri sunt de asemenea denumite coduri *prefix*). Fie l_i lungimea în biți a cuvântului de cod alocat simbolului numărul i din alfabet. O condiție necesară pentru cod de a fi fără prefix este ca $K = \sum_{i=1}^{n} 2^{-l_i} \leq 1$; aceasta este cunoscută ca *inegalitatea Kraft*. În același timp, pentru un cod de minimă redundanță, cantitatea $B = \sum_{i=1}^{n} l_i p_i$ trebuie să fie minimizată. Punctul de echilibru între aceste două cerințe – când K este cât mai mare posibil, dar nu mai mare de 1, iar B cât se poate de mic – este cel care definește codul de minimă redundanță.

A avea un set de lungimi ale cuvintelor de cod care satisfac inegalitatea Kraft nu este, prin ea însăși, o condiție suficientă pentru a garanta că acel cod este fără prefix. De exemplu, considerăm un alfabet de două simboluri, cu codurile „0" și „00". Suma Kraft, K, este 0,75, dar codul, evident, nu este fără prefix. În orice caz, dându-se o listă de lungimi de cod l_i care satisface inegalitatea Kraft, se poate întotdeauna proiecta un cod fără prefixe, cu cuvinte de cod de lungimile specificate. Pentru a vedea aceasta, presupunem că l_i este lista dată a lungimilor cuvintelor de cod și că $n_\ell = |\{i \mid l_i = \ell\}|$ este numărul de cuvinte de cod de lungime l, pentru $1 \leq l \leq L$, unde L este lungimea celui mai lung cuvânt de cod. Atunci, cel de-al $(j+1)$-lea din cele n_l cuvinte de cod de lungime l trebuie să fie un întreg binar de l biți, notat j + baza $[l]$, unde

$$\text{bază}[\ell] = \left\lceil \frac{\left(\sum_{k=\ell+1}^{L} n_k \cdot 2^{L-k}\right)}{2^{L-\ell}} \right\rceil .$$

[a] Într-adevăr, această întreagă clasă de codoare este uneori numită „codoare Huffman" din cauza contribuțiilor inițiale ale lui Huffman. Oricum, este de asemenea adevărat că nu toate codoarele de minimă redundanță pot fi generate de algoritmul lui Huffman și pentru acest motiv preferăm să folosim denumirea mai generală de coduri de „minimă redundanță".

De pildă, considerăm exemplul arătat în tabelul 1. Din lungimile l_i listate în cea de-a doua coloană, se poate vedea că $L = 5$, și că $n_5 = 2$, $n_4 = 3$, $n_3 = 0$, $n_2 = 3$ și $n_1 = 0$. Prin urmare, *baza* [5] = 0, *baza* [4] = 1, *baza* [3] = 2, *baza* [2] = 1 si *baza* [1] = 2. Cea de-a patra coloană listează valorile lui j corespunzătoare fiecărui simbol i, evidențiind (din zero) cuvintele de cod de o lungime dată. Întregii de l_i biți rezultând din *baza* $[l_i] + j$ sunt listați în coloana cea mai din dreapta; acestea sunt cuvintele de cod alocate.

Tabelul 1. Alocarea unui cod canonic

i	l_i	*bază*[l_i]	j	Cuvânt de cod
1	5	0	0	00000
2	5		1	00001
3	4	1	0	0001
4	4		1	0010
5	4		2	0011
6	2	1	0	01
7	2		1	10
8	2		2	11

Se observă un tipar (model) regulat – toate cuvintele de cod de o lungime dată sunt întregi binari consecutivi. Acest aranjament este cunoscut ca un cod *canonic*,[10-12] și permite codarea rapidă și decodarea utilizând exact două tabele de L cuvinte, unul pentru matricea *bază* și al doilea pentru a înregistra primul număr de simbol al fiecărei lungimi de cod. În particular, să notăm că nu este necesară întreținerea unei liste a tuturor cuvintelor de cod atât pentru codor cât și pentru decodor, astfel încât alfabete mari pot fi mânuite cu eficiență.

Pentru exemplul arătat în tabelul 1, al doilea tabel *offset* va înmagazina valorile *offset*[5] = 1, *offset* [4] = 3, *offset* [3] = 6, *offset*[2] = 6, si *offset*[1] = 9, indicând că primul cuvânt de cod de 5 biți este alocat simbolului 1, primul cuvânt de cod de 4 biți este alocat simbolului 3 și așa mai departe.

Fiind date aceste două tabele și o funcție *next_input_bit*() care întoarce următorul bit din fluxul de date comprimat, decodarea unui simbol este făcută de următorii pași:

1. Set *code* ← *next_input_bit*() și *length* ← 1.
2. While *code* < *base* [*length*] do
   Set code ← 2**code* + *next_input_bit*() *and length* ← *length* + 1.
3. Return *offset* [*length*] + *code* – *base* [*length*].

Bucla strânsă și tiparul de referință din memorie bine localizat înseamnă că se pot decoda foarte repede codurile canonice.[13,14] Se pot, de asemenea, decoda coduri de minimă redundanță, fără nici un fel de manipulare la nivel de bit,[15,16] deși resurse de memorie considerabile ar putea fi necesare pentru orice alfabet, cu excepția celor foarte mici.

Problema proiectării unui cod de minimă redundanță este prin urmare problema calculării lungimii care trebuie alocată fiecărui simbol. Cerința finală de a aloca de fapt cuvinte de cod nu trebuie lăsată să umbrească această problemă, iar în cele ce urmează în acest capitol vom presupune că generarea codurilor canonice se realizează automat

de îndată ce s-au calculat lungimile cuvintelor de cod și că problema principală este de a găsi aceste lungimi.

Să ne întoarcem acum la algoritmul Huffman. Fiecărui simbol i i se alocă o lungime de cod inițială de $l_i = 0$ și este plasat ca un element singular cu o pondere p_i într-o listă de *pachete*. Acest aranjament inițial pentru distribuția ponderilor P = [1, 2, 3, 4, 4, 10, 11, 12] este arătată în primul rând din tabelul 2. În tabelul 2 valorile îngroșate din fața fiecărui pachet de simboluri reprezintă ponderea combinată a acelui pachet. Apoi, în cadrul fiecărui pachet, lungimile de cod alocate până acum diferitelor simboluri sunt indicate prin lista de numere.

În fiecare etapă a algoritmului Huffman, pachetele cu cele mai mici două ponderi sunt identificate și combinate. Din aceasta rezultă un singur pachet mai mare, cu ponderea egală cu suma ponderilor celor două pachete constituente, cu fiecare lungime de cod din cadrul celor două pachete incrementate cu unu și cu cele două seturi de lungimi unite. De exemplu, cele două pachete combinate sunt **1**(l_1=0) și **2**(l_2=0); pachetul rezultant este **3**(l_1=1, l_2=1). Pentru motive ce vor fi explicate mai jos, separăm originalele de pachetele care rezulă din pașii combinărilor; ultimele sunt în al doilea rând în fiecare secțiune din tabelul 2.

La finalul algoritmului rămâne un singur pachet și acel pachet conduce la codurile de minimă redundanță. În exemplul din tabelul 2, lungimile de cod finale sunt acelea pe care le-am presupus mai înainte în tabelul 1. Ca urmare, o implementare a algoritmului Huffman ar trebui să fie de tip cod-arborescent, cu marginile arborelui notate cu „0" și „1" biți; apoi scăpăm de aceste cuvinte de cod prin traversarea acestui cod-arborescent. Oricum, procedând așa, ajungem la o alocare de cuvinte de cod care poate să nu se potrivească cu tiparul canonic cerut de o decodare rapidă, ghidată de un tabel. De exemplu, pentru distribuția de ponderi presupusă în tabelul 2 nu este nici o notare a marginilor arborelui, care să ducă la codul canonic arătat în tabelul 1. De aceea insistăm ca ieșirea dintr-un generator de cuvinte de cod să fie o listă de lungimi corespunzătoare cuvintelor de cod.

Tabelul 2 dorește să ilustreze modelul din spatele algoritmului Huffman, mai degrabă decât mecanismul prin care metoda trebuie executată. Implementarea reală se dovedește un exercițiu interesant de utilizare a structurii datelor și de proiectare a algoritmului. Van Leeuwen[17] a observat că dacă ponderile intrărilor sunt sortate, atunci menținerea a două liste de pachete (așa cum se arată în tabelul 2) permite ca generarea unui cod de minimă redundanță să fie realizată în timpul $O(n)$; adică, la un cost de calcul care crește liniar cu mărimea alfabetului. O astfel de eficiență este posibilă deoarece la fiecare etapă a procesului, cele două pachete cu ponderile cele mai mici vor fi între cele două prime pachete ale fiecăreia din cele două liste și pot astfel să fie identificate repede. Van Leeuwen a demonstrat, de asemenea, că pentru cazul în care ponderile nu sunt deja sortate, este cel mai bine să fie întâi sortate și abia apoi să se aplice metoda de mai sus. Adică, el a arătat că trebuie parcurși $O(n \log n)$ pași pentru a genera un cod de minimă redundanță pentru o distribuție de ponderi nesortate. De aceea, în acest capitol, ne vom focaliza pe distribuțiile sortate.

Tabelul 2. Calculul codului de minimă redundanță

	Liste
0	$1(l_1 = 0), 2(l_2 = 0), 3(l_3 = 0), 4(l_4 = 0), 4(l_5 = 0), 10(l_6 = 0), 11(l_7 = 0), 12(l_8 = 0)$
1	–
	$3(l_3 = 0), 4(l_4 = 0), 4(l_5 = 0), 10(l_6 = 0), 11(l_7 = 0), 12(l_8 = 0)$
	$3(l_1 = 1, l_2 = 1)$
2	$4(l_4 = 0), 4(l_5 = 0), 10(l_6 = 0), 11(l_7 = 0), 12(l_8 = 0)$
	$6(l_1 = 2, l_2 = 2, l_3 = 1),$
3	$10(l_6 = 0), 11(l_7 = 0), 12(l_8 = 0)$
	$6(l_1 = 2, l_2 = 2, l_3 = 1), 8(l_4 = 1, l_5 = 1)$
4	$10(l_6 = 0), 11(l_7 = 0), 12(l_8 = 0)$
	$14(l_1 = 3, l_2 = 3, l_3 = 2, l_4 = 2, l_5 = 2)$
5	$12(l_8 = 0)$
	$14(l_1 = 3, l_2 = 3, l_3 = 2, l_4 = 2, l_5 = 2), 21(l_6 = 1, l_7 = 1),$
6	–
	$21(l_6 = 1, l_7 = 1), 26(l_1 = 4, l_2 = 4, l_3 = 3, l_4 = 3, l_5 = 3, l_8 = 1)$
7	–
	$47(l_1 = 5, l_2 = 5, l_3 = 4, l_4 = 4, l_5 = 4, l_6 = 2, l_7 = 2, l_8 = 2)$

Moffat și Katajainen[18] au rafinat mai departe abordarea lui Leeuwen. Implementarea lor nu numai că execută operația într-un timp $O(n)$ pentru o intrare sortată, ci, oricât de surprinzător ar părea, nu cere aproape nici un spațiu de memorie suplimentară cu excepția spațiului ocupat de tabelul ponderilor simbolurilor de intrare. Dat fiind un tabel sortat de n ponderi de simboluri, această metodă *in-situ* operează în timp liniar după n și înlocuiește frecvența fiecărui simbol cu lungimea în biți a cuvintelor corespunzătoare de cod de minimă redundanță, folosind doar un mic număr constant de locații suplimentare de memorie. Metoda este foarte rapidă în practică, iar Moffat și Katajainen[18] au oferit rezultate demonstrând calcularea unui cod de minimă redundanță, pentru un alfabet de peste un milion de simboluri, în mai puțin de două secunde.

Este de asemenea necesar să arătăm pe baza reducerii la absurd, că este necesar un timp cel puțin liniar (de n) pentru dezvoltarea unui cod de minimă redundanță, chiar dacă ponderile simbolurilor sunt prezentate în formă ordonată. Fie n ponderi de simboluri date de $p_i = 3i$, pentru $1 \le i \le n-2$ și $p_{n-1} = p_n = \left(\sum_{i=1}^{n-2} p_i\right) - 1$. Date fiind aceste ponderi, fiecare algoritm care rezolvă această problemă trebuie să aloce $l_{n-1} = l_n = 2$, căci dacă nu ar fi așa, o simplă rearanjare a cuvintelor de cod ar conduce la o codare mai eficientă. Să presupunem atunci că un algoritm nu a analizat toate cele n valori p_i, ceea ce ar fi exact cazul în care ar opera într-un timp subliniar de n. Fie p_j o astfel de valoare neexaminată. Atunci, cu condiția menținerii inegalității $p_{j-1} \le p_j \le p_{j+1}$, am putea să realocăm p_j, de vreme ce algoritmul nu a luat în considerare tocmai valoarea lui p_j. Să presupunem că am atribui $p_j = 3j\text{-}2$. Acum, codul de minimă redundanță ar trebui să aibă fie $l_{n-1} = 1$, fie $l_n = 1$, de vreme ce, din nou, dacă nu ar fi așa, o rearanjare simplă ar reduce costurile. Dar, dacă un algoritm nu îl analizează pe p_j, atunci el nu este capabil să distingă între cazurile $p_j = 3j$ și $p_j = 3j - 2$ și rezultă că nu poate construi totdeauna coduri de minimă redundanță. Prin urmare, fiecare simbol p_i trebuie analizat și fiecare algoritm trebuie să consume un timp $O(n)$.

Date fiind această limită inferioară, metoda de calcul Moffat și Katajainen[18] și folosirea alocării canonice a cuvintelor de cod problema codării de minimă redundanță este în esență rezolvată.

Sunt, de altfel, o varietate de probleme legate de construcția codurilor pentru care situația nu este chiar atât de clară. Un caz important special este problema codării de *lungime limitată*, în care trebuie realizat un cod de minimă redundanță cu condiția suplimentară ca $l_i \leq L$, pentru o limită specificată L. O aplicație practică a acestui stil de codare este accea când, din motive de viteză, o implementare presupune că toate cuvintele de cod pot fi manipulate ca întregi pe calculatorul pe care sunt folosite (cum a fost cazul, de exemplu, în pseudo-codurile decodării canonice date mai înainte). Abia în 1990 s-a dat o soluție eficientă problemei codării cu lungime limitată, prin algoritmul *pachetelor-sudate* descris de Larmore și Hirschberg[19]. Metoda cere un timp $O(nL)$ și un spațiu $O(n)$ și cu toate că se bazează pe aceeași paradigmă ca și algoritmul Huffman, ea reprezintă un ordin de micșorare a magnitudinii (*an order of magnitude slower*). Katajainen *et al.*[20] au rafinat implementarea algoritmului pachetelor-sudate și au arătat că surplusul de spațiu se poate reduce la $O(L^2)$. În practică L este o constantă mică, de exemplu 32 și astfel această din urmă metodă poate fi privită ca fiind „aproape la locul ei". Această metodă de *graniță a pachetelor sudate* este totodată suficient de rapidă ca să fie utilizată în sisteme practice de compresie. Katajainen *et al.* descriu experimente în care un cod cu lungime limitată pentru un alfabet de peste un milion de simboluri a fost construit în circa 180 secunde de timp CPU. Codul limitat la 4 biți pentru exemplul arătat în tabelul 2 este $l = [4, 4, 4, 4, 3, 3, 2, 2]$, cu costul $B = 128$ biți; costul codului de minimă redundanță este 125 biți.

Algoritmi asimptotic mai rapizi au fost dezvoltați pentru problema codării de lungime limitată (de exemplu vezi Scheiber[21] și bibliografia inclusă aici). Deși operează în limite de timp sub $O(nL)$ pentru anumite combinații de L și n, nu pare de loc potrivită utilizarea acestor algoritmi pentru scopuri practice. Dezvoltarea lor este interesantă dintr-un punct de vedere teoretic – în contrast cu problema lungimii nelimitate, nu a fost dedusă nici o limită inferioară ne-banală a problemei complexității codării de lungime limitată – dar au puține șanse de a fi folosite în practică.

Moffat și Turpin[22] consideră un alt mecanism prin care redundanța minimă și limitarea lungimii pot fi specificate. Ei au observat că pentru alfabete mari, lista P care descrie ponderile alfabetului conține de obicei multe elemente care se repetă, și că o reprezentare mult mai succintă este înregistrarea numărului de simboluri pentru fiecare pondere dată. Următoarea observație este că dacă sunt r simboluri distincte de pondere w_i și dacă fiecare din aceste ponderi apare de f_i ori (astfel că $\sum_{i=1}^{r} f_i = n$), atunci codul de minimă redundanță poate fi calculat într-un timp și un spațiu $O(r \log (n/r))$, care este subliniar, când r este subliniar în n. (De notat că aceasta nu contrazice demonstrația prin reducere la absurd de mai sus, de vreme ce demonstrația se baza pe faptul că toate p_i sunt distincte.) Moffat și Turpin[22] au arătat, de asemenea, cum aceleași idei pot fi aplicate la producerea unui cod de lungime limitată, dând un algoritm $O(Lr \log (n/r))$. Lista celor 1.000.000 de frecvențe ale cuvintelor menționate mai sus are r = 11.000 frecvențe distincte (într-adevăr, principiul porumbarului conduce la rezultatul că un flux de N simboluri poate avea cel mult $r = \sqrt{2N}$ frecvențe distincte ale simbolurilor și pentru aceste date N = aprox. 500.000.000) și un cod de lungime limitată poate fi calculat în numai câteva secunde.

O altă variantă a codării de minimă redundanță este *codarea alfabetică*. Condiția suplimentară impusă în această situație este ca toate cuvintele de cod să fie ordonate din punct de vedere lexicografic în același mod ca și simbolurile originale. Aceasta permite sortarea șirurilor care au fost comprimate folosind alfabetul, fără să fie necesară decomprimarea lor. Acesta este un caz în care codurile canonice nu pot fi folosite, deoarece nu este posibil să presupunem ca intrare o listă ordonată a ponderilor simbolurilor. Algoritmul clasic pentru această problemă este datorat lui Hu și Tucker[23]; din nou, au existat cercetări recente în această zonă și Larmore și Przytycka[24] consideră împreună situațiile când este impusă limitarea lungimii și codul trebuie să fie alfabetic. Zandi, Iyer și Langdon[25] consideră cazul când sunt reprezentate expresii multi-simbol. Tabelul 3 dă un exemplu de cod alfabetic de minimă redundanță pentru un alfabet de 6 simboluri. Acest cod costă $B = 80$ de biți în comparație cu un cod de minimă redundanță la care $B = 68$ biți.

Tabelul 3. Exemplu de cod alfabetic de minimă redundanță

Simbol	Pondere	Cuvânt de cod
A	2	00
B	10	01
C	16	10
D	2	1100
E	1	1101
F	3	111

În final, să notăm că Larmore și Przytycka[24] au dezvoltat primul algoritm paralel pentru codarea de minimă redundanță, care necesită un timp sub $O(n^2)$ (timp multiplicat de cerințele procesorului). Această metodă se bazează pe o paradigmă complet nouă pentru calcularea codurilor de minimă redundanță și deschide calea pentru îmbunătățiri ulterioare ale algoritmilor pentru codurile de lungime limitată și codarea paralelă de minimă redundanță.

2.6. *Codarea aritmetică*

Dat fiind faptul că bitul este unitatea de stocare a datelor, pare imposibil ca un cuvânt de cod să ocupe fracțiuni de bit. Și dat fiind că un cod de minimă redundanță este cel mai bun cod care se poate obține folosind cuvinte de cod de lungime integrală, ar părea că un cod de minimă redundanță este *optimal*, adică, este cel mai apropiat de entropie posibil.

Surprinzător, deși adevărată pentru codarea unui singur simbol, acest raționament nu este valabil când trebuie codate șiruri de simboluri. Un singur cod trebuie să fie clar un număr întreg de biți. Dar să presupunem că trebuie prezentat un șir de simboluri. Cu condiția ca întregul mesaj să fie un număr întreg de biți, nu există nici o restricție ca fiecare bit din forma codată să fie alocat exclusiv unui singur simbol sau unui altuia. De exemplu, dacă cinci simboluri echi-probabile sunt reprezentate cumva într-un total de trei biți, atunci nu este nerezonabil să simplificăm situația și să afirmăm că fiecare simbol ocupă 0,6 biți. Ieșirea trebuie să fie evident „cu cocoloașe" (întreruperi și aglomerări), deoarece biții de ieșire pot fi emiși numai după, să zicem, al doilea, al patrulea și al cincilea bit din acest șir. Oricum, dacă există o *stare internă* a codorului

care este parte a procesului de compresie și dacă după ce fiecare simbol este codat starea se poate modifica, atunci codul total pentru fiecare simbol este numărul de biți de la ieșire plus numărul „potențial" al modificării de stare internă. Deoarece aceasta din urmă poate fi fracțională într-un fel, este chiar de dorit ca un codor să reprezinte un simbol de probabilitate p_i în $-\log_2 p_i$ biți. La sfârșitul șirului starea internă trebuie, desigur, să fie reprezentată într-un fel; de fapt, prin rotunjirea la un număr întreg de biți.

Codarea aritmetică[26-29] este un mecanism care face tocmai acest lucru. Starea internă a codorului este înregistrată folosind două variabile L și R, înregistrând limita de jos a unui interval mărginit (L – de la *low*) și gama acelui interval (R – de la *range*). Pentru scopul acestei discuții, vom presupune că aceste valori sunt numere reale cuprinse între zero și unu. În implementări ele sunt de fapt întregi, scalați cu o putere corespunzătoare a lui doi.

Inițial $L = 0$ și $R = 1$. *Potențialul* intern al codorului este $-\log_2 R$. Presupunând că $P = [p_i]$ este o distribuție de probabilitate normalizată, astfel încât $\sum_{i=1}^{n} p_i = 1$. Definim $low[i] = \sum_{j=1}^{i-1} p_j$, probabilitatea cumulativă a tuturor simbolurilor dinaintea lui i din alfabet. Este, de asemenea, util să presupunem că este definit $low\,[n + 1] = 1$. Atunci pentru a coda simbolul numărul i din alfabet se execută următorii doi pași:

1. Set $L \leftarrow L + R \times low\,[i]$.
2. Set $R \leftarrow R \times p_i$.

La sfârșitul șirului, codul transmis este orice număr c astfel încât $L \le c < L + R$. În același timp, $R = \prod_{k=1}^{m} p_{s_k}$, unde s_k este simbolul numărul k de la intrare și sunt m simboluri în șir. Potențialul a crescut similar la $-\sum_{k=1}^{m} \log_2 p_{s_k}$ și pentru a garanta că el este în gama specificată, numărul c trebuie să aibă lungimea cel puțin egală cu acest număr de biți. De exemplu, să presupunem că un alfabet are $n = 2$ simboluri și că $p_1 = 3/4$ și $p_2 = 1/4$. Atunci o secvență cu $m = 6$ și $s_1 = s_2 = s_3 = s_4 = s_5 = 1$ și $s_6 = 2$ conduce la $L = 729 / 4096 = 0{,}17798$ și $R = 243 / 4096 = 0{,}05933$. Aceasta conduce în binar la $L = 0.001011011001_2$ și $L + R = 0.001111001100_2$, deci un mesaj șir neambiguu între aceste două limite este $c = 0{,}00110$, adică, șirul „00110". În acest caz, cinci biți ajung pentru a reprezenta cele șase simboluri. Aceasta se poate verifica: $\lceil -\log_2((3/4)^5(1/4)) \rceil = \lceil 4{,}08 \rceil = 5$.

Pentru a realiza o implementare care să funcționeze, trebuie rezolvate câteva probleme. Cea mai importantă dintre acestea este că procesul tocmai descris necesită precizie aritmetică arbitrară. Dacă fișierul de ieșire este de (să zicem) 125 kb, atunci L și R trebuie menținute la o precizie de aproximativ un milion de biți, o imposibilitate substanțială. Din fericire, este de asemenea posibil să creăm biți pe baza incrementării și astfel să limităm precizia necesară pentru L și R. După fiecare pas potențial de incrementare (mai sus), trebuie executată următoarea *re-normalizare* (sau pasul de reducere a potențialului):

```
3. Cand R < 0.25 executa:
a) Daca L + R < 0.5
     /* primul bit din toate valorile c astfel incat L ≤ c < L + R este un „0" */
     bit_plus_follow(0)
b) In plus, daca 0.5 ≤ L atunci
```

```
    /* primul bit din toate valorile c a.i. L ≤ c < L + R este un „1“ */
    bit_plus_follow(1)
    Set L ← L – 0.5

c)  In plus,
    /* Polaritatea bitului este opusa urmatorului bit */
    Set bits_outstanding ← bits_outstanding + 1,
    Set L ← L – 0.25

d)  Atunci, pentru toate cazurile set L ← 2 x L and R ← 2 x R
```

4. Pentru a calcula *bit_plus_follow(b)*, executa:
   a) *write_one_bit(b)*
   b) Foloseste *write_one_bit*(1 – *b*) pentru a tipari *bits_outstanding* biti de polaritate opusa
   c) Set *bits_outstanding* ← 0.

Acest mecanism menține $R \geq 0{,}25$ înainte de codarea fiecărui simbol și astfel precizia necesară pentru L și R este limitată la doi biți în plus față de precizia cu care sunt păstrate probabilitățile p_i.

Figura 2 arată trei alternative oferite pentru pașii 0a, 0b, respectiv 0c. În primul caz, (figura 2(a)) bitul următor de ieșire este clar un zero și valorile pentru L și R pot fi ajustate în consecință. Al doilea caz (figura 2(b)) se adresează situației în care bitul următor este un unu. Al treilea caz, la pasul 0c și arătat în figura 2(c), este într-un fel mai complex. Când $R < 0{,}25$ și L și $L+R$ sunt de cealaltă parte a lui 0,5, polaritatea următorului bit de ieșire nu poate fi cunoscută încă, de vreme ce depinde de simboluri următoare care nu au fost codate încă. Ceea ce se poate cunoaște, oricum, este că bitul *de după* următorul bit va fi de polaritate diferită față de a următorului bit, de vreme ce toate numerele binare din plaja 0,25 – 0,75 încep fie cu „01“ fie cu „10“. Prin urmare, în acest al treilea caz, renormalizarea – dublarea lui R și L – poate avea loc, cu condiția de a introduce o notă (folosind *bits_outstanding*) pentru a scoate un bit suplimentar opus la ieșire atunci când urmează a se produce un bit de polaritate definită. Acesta este motivul pentru care de fiecare dată când este scos la ieșire un bit la pasul 0c, este urmat de orice *bits_outstanding* încă existent.

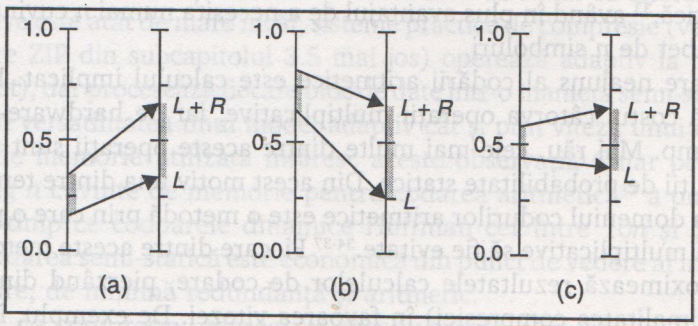


Figura 2. Renormalizarea în codarea aritmetică

Decodorul trebuie să inverseze acest proces. Dat fiind un cod c, decodorul trebuie să determine secvența celor m simboluri care au condus la transmiterea lui c. Aceasta

se face prin metoda următoare. Se presupune cantitatea V ca fiind fereastra curentă în c, de aceeași precizie ca aceea folosită pentru L și R. Limitele lui L și R sunt din nou inițializate la 0 și respectiv 1 înainte de primul simbol, iar V trebuie inițializat la numărul corespunzător de biți de la începutul șirului de biți c.

1. Se determină i astfel încât $low\ [i] \leq (V - R)/R < low\ [i + 1]$
2. Set $L \leftarrow L + R \times low\ [i]$.
3. Set $R \leftarrow R \times p_i$.
4. Se renormalizează ca în pasul 0c de la codor, tratând pe V ca și pe L, cu excepția faptului că de fiecare dată când L este dublat, V trebuie dublat și un bit din șirul codat de biți adăugat într-o poziție de ordin inferior, folosind *next_input_bit*().
5. Se scoate la ieșire simbolul i.

Într-o implementare practică pentru folosirea împreună cu un model adaptiv, probabilitățile sunt supuse unei revizuiri continue, astfel că probabilitățile cumulative presupuse în zona *low* nu pot fi calculate în avans. Nici nu pot fi presupuse a fi normalizate. În schimb, trebuie să fie calculate și actualizate din mers. Normalizarea poate fi realizată cu prețul unor divizări suplimentare în timpul procesului de codare; dacă c_i este numărul simbolului i într-un șir de m simboluri, atunci $p_i = c_i / m$ este gata calculat.

Problema menținerii eficiente a numărărilor frecvenței cumulative, pentru a stăpâni un model adaptiv, este și mai provocatoare. Cu un alfabet de n simboluri, ar fi necesar un timp $O(n)$ dacă *low* [i] ar trebui calculat ca nou la fiecare pas de codare. În schimb, s-au propus mai multe structuri care să permită calcularea lui *low[i]* într-un timp subliniar. Acestea includ o organizare cu revenire la începutul listei, descrisă de Witten *et al.*,[29] abordarea arborescentă plată (*splay-tree*) a lui Jones,[30] abordarea arborescentă întoarsă (*skewed-tree*) a lui Moffat[31] (care are o structură de acces analogă codului Elias, C_γ, discutat anterior) și metoda arborescentă echilibrată (*balanced tree*) a lui Fenwick.[32]

Dintre acestea, numai tehnicile *splay-tree* și *skewed-tree* sunt liniare în timp. Folosind oricare dintre aceste structuri, toate operațiile necesare pot fi făcute într-un timp $O(m + b)$ unde b este numărul de biți produși de codor și m este numărul de simboluri codate. Metoda *balanced-tree*,[32] sub-liniară în situații extreme, este de asemenea foarte rapidă în practică,[33] având în plus avantajul de a necesita numai n cuvinte de memorie pentru un alfabet de n simboluri.

Cel mai mare neajuns al codării aritmetice este calculul implicat. Fiecare simbol codat incumbă costul câtorva operații multiplicative, iar pe hardware-ul uzual acest lucru este scump. Mai rău, cele mai multe dintre aceste operații sunt necesare chiar pentru distribuții de probabilitate statice. Din acest motiv una dintre temele principale de cercetare în domeniul codurilor aritmetice este o metodă prin care o parte sau toate aceste operații multiplicative să fie evitate.[34-37] Fiecare dintre aceste metode, într-un fel sau altul, aproximează rezultatele calculelor de codare, pierzând din exactitate (și astfel, din optimalitatea compresiei) în favoarea vitezei. De exemplu, în unele dintre aceste metode, R este aproximat cu un număr $R' \leq R$ de o formă specială și multiplicările și divizările cu R sunt aproximate cu multiplicări și divizări cu R', implementate ca un număr mic de operații de adunare și deplasare.

2.7. *Performanțe relative*

Cititorul perspicace va fi observat că ne-am focalizat mai mult pe metodele de codare statică atunci când am discutat codurile de minimă redundanță, dar ne-am concentrat mai mult pe codarea adaptivă când am discutat despre codarea aritmetică. Nu a fost accidental. În mod clar, codarea aritmetică poate fi, de asemenea, folosită pentru distribuții de probabilitate statice; și totodată, o varietate de metode „Huffman dinamice" au fost descrise în literatură.[38-42] Oricum, codarea statică aritmetică nu este mult mai rapidă decât codarea dinamică aritmetică, în timp ce codarea dinamică Huffman este substanțial mai lentă decât codarea statică (canonică) Huffman. Figura 3, derivată din rezultatele prezentate de Muffat *et al.*[33], arată vitezele relative ale unor metode de codare descrise în acest capitol. Datele de test folosite au fost o listă de 1.000.000 de întregi corespunzând unor cuvinte distincte dintr-o mostră de text în limba engleză.

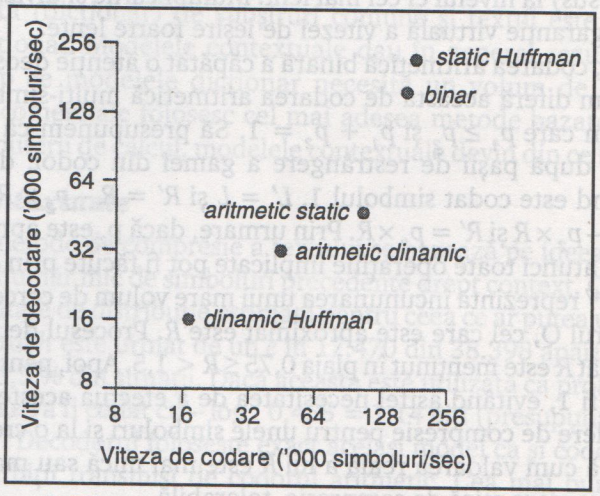


Figura 3. Viteza de codare, mii de simboluri pe secundă

Concluzia ce se poate trage din figura 3 este clară: dacă este cerut un codor dinamic, va fi preferată codarea aritmetică; dar pentru codarea statică, trebuie folosit un codor canonic de minimă redundanță. Merită de asemenea de notat că diferența între aceste două alternative este atât de mare încât, sisteme practice de compresie (vezi, de exemplu, discuția despre ZIP din subcapitolul 3.5 mai jos) operează adaptiv la nivel de blocuri (multi-kilooctet), dar procesează fiecare bloc de date într-o manieră semi-statică, câștigând astfel atât prin versatilitatea unui model adaptiv cât și prin viteza unui codor static.

Cerințele de memorie utilizată întăresc aceste observații. Chiar pentru un model dinamic, ajung n cuvinte de memorie pentru codarea aritmetică[32] a unui alfabet de n simboluri, în timp ce codoarele dinamice Huffman cer între $10n$ și $15n$ cuvinte de memorie.[33] Codarea semi-statică este economică din punct de vedere al memoriei pentru ambele codoare, de minimă redundanță și aritmetic.

În termeni de compresie, adesea nu prea avem de ales între cele două clase de codoare entropice, mai ales dacă alfabetul este mare și simbolurile cu probabilitatea cea mai mare au frecvență relativ mică. În principiu, codarea aritmetică nu va da niciodată o compresie mai slabă decât codurile de minimă redundanță, dar din cauza surplusului

din terminație și a necesității de a specifica mai precis probabilitățile simbolurilor, există o serie de situații practice în care un cod de minimă redundanță în stilul Huffman poate să conducă la o compresie ușor mai bună decât un cod aritmetic.[33,43] Mai mulți autori au remarcat limitele analitice ale ineficienței codurilor de minimă redundanță și au arătat că sunt foarte precise pentru cele mai mult aplicații.[38,44-46]

2.8. *Codarea alfabetelor binare*

Un domeniu important în care codarea aritmetică este esențială este codarea alfabetelor binare și metodele folosite pentru această problemă merită o mențiune specială. Când alfabetul de intrare este binar – pixeli într-o imagine cu două niveluri, sau biți într-un fișier de date text – singura metodă de codare care poate obține o compresie este codarea aritmetică. Dar acest mediu conduce codarea convențională aritmetică (așa cum a fost descrisă mai sus) la nivelul ei cel mai lent. Multiplicările și divizările bazate pe biți-per-intrare sunt o garanție virtuală a vitezei de ieșire foarte lente.

Din acest motiv, codarea aritmetică binară a căpătat o atenție deosebită în literatură. Pentru a vedea cum diferă aceasta de codarea aritmetică multi-simbol, considerăm ca exemplu situația în care $p_1 \geq p_2$ și $p_1 + p_2 = 1$. Să presupunem că L' și R' reprezintă valorile lui L și R după pașii de restrângere a gamei din codor, dar înaitea oricărei renormalizări. Când este codat simbolul 1, $L' = L$ și $R' = R - p_2 \times R$. Când se codează simbolul 2, $L' = R - p_2 \times R$ și $R' = p_2 \times R$. Prin urmare, dacă p_2 este aproximat cu o putere negativă a lui doi, atunci toate operațiile implicate pot fi făcute prin deplasare.

Codorul IBM Q[47] reprezintă încununarea unui mare volum de cercetări asupra acestui domeniu. În codorul Q, cel care este aproximat este R. Procesul de renormalizare este organizat astfel încât R este menținut în plaja $0{,}75 \leq R < 1{,}5$. Apoi, pentru toate înmulțirile, R este presupus a fi 1, evitând astfel necesitatea de a efectua aceste înmulțiri. Aceasta conduce la o pierdere de compresie pentru unele simboluri și la o creștere a compresiei pentru altele, după cum valoarea reală a lui R este mai mică sau mai mare decât 1. Pe ansamblu, este o pierdere mică de compresie, tolerabilă.

Codorul Q încorporează și un regim elegant de estimare a probabilităților.[48] În loc de a înregistra frecvența simbolurilor c_i și de a folosi c_i / m pentru a estima p_i, codorul Q își revizuiește estimarea numai când se scoate un bit la ieșire – adică, atunci când bucla de renormalizare operează. Notăm MPS (*Most Probable Symbol*) – simbolul cel mai probabil și LPS (*Less Probable Symbol*) – simbolul cel mai puțin probabil. Dacă simbolul care a cauzat bitul de la ieșire se întâmplă să fie MPS curent, probabilitatea lui MPS crește. Pe de altă parte, dacă bitul a fost cauzat de un LPS, crește probabilitatea lui LPS. Dacă cele două probabilități sunt presupuse egale, simbolul care a cauzat bitul de ieșire se consideră a fi MPS pentru următoarea secvență de pași de compresie. Valorile de probabilitate sunt restricționate la un număr redus de valori permise și toate schimbările de probabilitate se fac pe baza unui tabel director. În total, fiecare „context" de codor Q (mai precis, codor QM) necesită exact opt biți pentru stocare: unul indicând care simbol este MPS curent și șapte pentru a înregistra un index care notează probabilitatea curentă estimată. Prin evitarea divizării inerente estimărilor de probabilitate bazate pe frecvență și prin ajustarea estimărilor de probabilitate numai o dată per bit (ceea ce, pentru un codor binar, poate însemna un număr foarte mare de simboluri dacă compresia este bună), codorul Q oferă o compresie bună la viteze mari.

3. Modele pentru text

Compresia de text este obținută prin formarea unui model al textului care urmează a fi comprimat și folosirea acestui model pentru a furniza distribuții de probabilitate unui codor, așa cum se arată în figura 1. În capitolul precedent am discutat problema codării și am arătat cum este posibil să obținem o compresie optimală pentru o distribuție de probabilitate dată, folosind metoda codării aritmetice. Prin contrast, dezvoltarea metodelor de codare este oarecum o artă, mergând pe idei din lingvistică și statistică și cercetări pentru îmbunătățirea compresiei s-au orientat către proiectarea unor modele mai bune pentru text.

Sunt două abordări generale ale modelării textului. Una este de a folosi un model *contextual*, în care distribuția de probabilitate pentru fiecare simbol este estimată pe baza contextului în care acel simbol a apărut. Cealaltă este un model dicționar, în care este păstrată o listă (dicționar) de subșiruri comune și textul este codat substituind referințele din dicționar. Modelele contextuale dau în general rezultate mai bune de compresie, în timp ce modelele dicționar necesită un volum de calcul mai redus. Sistemele curente comerciale folosesc cel mai adesea metode bazate pe dicționar, dar din cauza creșterii puterii de calcul, modelele contextuale devin din ce în ce mai atractive.

3.1. *Modele contextuale*

Cele mai bune metode de compresie a textului se bazează pe ideea de a „prezice" un simbol folosind un număr mic de simboluri precedente drept context. O predicție este pur și simplu o probabilitate de distribuție estimată pentru ceea ce ar putea urma. De exemplu, într-un text, contextul *h* este urmat de un *e* în 17.470 din 38.398 apariții ale contextului, cu alte cuvinte în 45,5 % din situații. Dacă aceasta este utilizată ca probabilitate estimată pentru literă, atunci *e* va fi codat cu $-\log_2 0{,}455 = 1{,}14$ biți (presupunând că se folosește un codor aritmetic). Decodorul folosește exact același model ca și codorul și poate astfel să refacă litera din biții transmiși de codorul aritmetic. Cea mai bună compresie este obținută dacă litera care apare de fapt în context are o probabilitate mare. Arta în cazul modelării este de a face ca aceasta să se întâmple cât mai des.

O cale de a face predicții mai exacte este de a folosi contexte mai mari. De exemplu, în contextul a două caractere *th*, un *e* apare în 56,5% din cazuri și, folosind acest model va fi codat cu 0,82 biți.

Un model care face predicții bazate pe n simboluri precedente este numit un model contextual finit de ordinul n, unde $n \geq 0$. Chiar cazul banal al unui context de ordinul zero poate oferi o compresie rezonabilă în unele circumstanțe, dar de regulă sunt necesare contexte de ordine superioare pentru a obține o bună compresie. Ar părea de dorit să se stabilească n cât se poate de mare. Oricum, un context prea mare devine improbabil să fi fost văzut vreodată în eșantionul de text pe care se bazează distribuția de probabilitate. De asemenea, probabilitățile de predicție vor depinde foarte mult de eșantionul particular de text. În practică, cea mai bună abordare este de a folosi o varietate de dimensiuni ale contextului și un model care comută între acestea în funcție de nivelul de încredere al predicției corespunzătoare contextelor. Una dintre cele mai bune tehnici de compresie de text, metoda PPM (descrisă mai jos), folosește acest principiu.

Descrierea de mai sus a ignorat problema găsirii unui eșantion de text potrivit pe baza căruia să se facă estimarea statistică. Există trei posibilități: modelul *static, semi-*

static și *adaptiv*. În modelul *static*, statisticile se măsoară folosind unele mostre reprezentative de text și apoi rămân fixe pentru orice date viitoare ce vor fi comprimate. Această abordare nu este folosită larg în practică deoarece nu este practic posibil să ne asigurăm că mostra este reprezentativă pentru orice date care vor fi comprimate în viitor. Un *model semi-static* evită această problemă prin estimarea statisticilor din datele care se comprimă. Aceasta necesită două treceri prin date, o dată pentru a spicui statisticile și a doua oară pentru a coda datele. Se cere, de asemenea, ca modelul să fie memorat sau transmis o dată cu textul. Acest fel de model este potrivit când textul însuși este static. De exemplu, un volum mare de text care este stocat pe CD-ROM poate fi comprimat foarte eficient prin formarea unui model semi-static de ordin zero al cuvintelor din text.[13]

Sistemul de compresie cel mai versatil folosește *modelarea adaptivă*, unde statisticile se estimează din datele care au fost deja codate. Decodorul este capabil să calculeze estimări identice cu codorul deoarece va avea de decodat aceleași simboluri. În practică, modelul este de obicei actualizat imediat după codarea sau decodarea fiecărui simbol, decât să fie refăcut de la zero de fiecare dată. Un sistem adaptiv va da o compresie slabă la începutul fișierului deoarece statisticile se bazează pe un eșantion foarte mic. Oricum, mai târziu pe parcursul fișierului, compresia se îmbunătățește masiv, ajungând ca surplusul de la început să fie în mod tipic similar cu costul unui model care ar fi fost format într-o stare semi-statică.[49] Așadar, modelarea adaptivă dă performanțe de compresie similare cu modelarea semi-statică, dar parcurge o singură dată datele de intrare.

Modelele bazate pe context sunt de obicei legate de un codor entropic. Pentru motive discutate în ultimul capitol, un codor aritmetic este mai potrivit pentru modelarea adaptivă. Codarea de minimă redundanță este în general mai potrivită modelelor semi-statice, deoarece codul este generat numai o dată pentru fiecare fișier comprimat, în comparație cu un model adaptiv, care poate cere generarea de noi coduri de fiecare dată când statisticile se schimbă. Codarea de minimă redundanță este în particular potrivită pentru modelul de ordin zero bazat pe cuvinte, menționat mai sus, deoarece modelul conține probabilități relativ mici și astfel codorul comprimă la rate apropiate de entropia alfabetului.

3.2. *Modele dicționar*

Modelele dicționar folosesc o paradigmă în care câteva simboluri consecutive sunt înlocuite cu un singur cod. „Dicționarul" este o listă de fraze disponibile pentru substituție. Aceste fraze sunt în general subșiruri comune și construcția dicționarului poate fi statică, semi-statică sau adaptivă.

Cele mai larg folosite metode de compresie cu dicționar sunt de departe cele adaptive. Într-o situație adaptivă, dicționarul este construit folosind textul precedent codat. Inițial dicționarul este gol, sau conține unele subșiruri banale cum ar fi simbolurile alfabetului de intrare. Pe măsură ce textul este comprimat, șiruri „folositoare" sunt adăugate dicționarului și, în unele cazuri, intrări care nu sunt susceptibile a fi productive sunt eliminate.

Aproape toate metodele adaptive de compresie cu dicționar stau sub umbrela generală a codurilor „Ziv-Lempel", o familie de metode care derivă din două lucrări – germeni, de J. Ziv și A. Lempel.[50,51] Ambele lucrări descriu metode adaptive cu dicționare, deși tehnicile au diferențe semnificative. Ele sunt uneori etichetate LZ77 și LZ78, sau LZ1, respectiv LZ2.

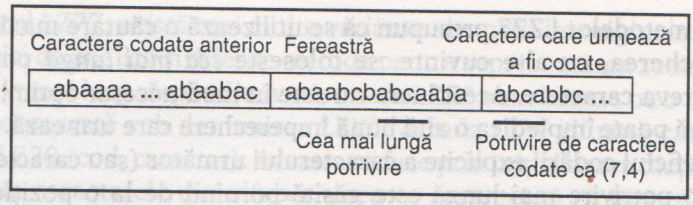


Figura 4. Un pas de codare pentru metoda de compresie LZ77

Familia de metode LZ77 foloseşte ca dicţionar o fereastră glisantă de simboluri recent-văzute, aşa cum este ilustrat în figura 4. Fereastra păstrează tipic între 1 şi 8 KB de caractere imediat precedente punctului curent de codare. Codorul caută în fereastră potrivirea cea mai lungă pentru următoarele câteva caractere. Potrivirea este codată prin transmiterea distanţei (*offset*) între punctul de codare curent şi locul unde începe potrivirea şi, de asemenea, lungimea potrivirii. De exemplu, în figura 4 o potrivire pentru patru caractere a fost găsită, începând cu şapte caractere înaintea poziţiei codate curent, astfel încât se va transmite perechea *pointer* (7,4) reprezentând offsetul şi lungimea. Caracterele codate sunt apoi adăugate ferestrei şi acelaşi număr de caractere „vechi" sunt scoase din celălalt capăt al ferestrei.

Decodarea este foarte eficientă pentru metoda LZ77, deoarece decodorul se uită pur şi simplu la sub-şirul potrivit din propria lui fereastră şi copiază caracterele indicate. Aceasta face tehnica foarte potrivită pentru situaţiile de „distribuţie", unde fişierele sunt codate o dată, distribuite în multe locuri şi decodate de multe ori.

Doi dintre factorii care disting metodele membre ale familiei LZ77, una de cealaltă, sunt cum se codează pointerul (codul indicator) şi ce se întâmplă dacă nu se găseşte nici o potrivire corespunzătoare. O abordare simplă de codare este să se reprezinte pointerul într-o fereastră de dimensiune N folosind $\lceil \log_2 N \rceil$ biţi. Oricum, îmbunătăţiri considerabile se pot aduce compresiei prin folosirea unor coduri scurte pentru pointere ale caracterelor mai recente şi coduri mai lungi pentru pointere corespunzătoare unor caractere aflate mult înapoi în fereastră. În acelaşi fel, lungimea potrivirii poate fi codată stabilind o lungime maximă F şi codând lungimea în $\lceil \log_2 F \rceil$ biţi, dar potriviri scurte sunt mult mai comune decât cele lungi şi o compresie mai bună se poate obţine prin varierea lungimii codului.

Uneori, mai ales la începutul codării, nu va fi găsită nici o potrivire pentru caracterele care trebuie codate, sau potrivirea este atât de mică încât pointerul ar consuma mai mult spaţiu decât caracterele pe care le înlocuieşte. Metodele LZ77 necesită în mod tipic o minimă potrivire pe lungime de două sau trei caractere pentru a merita codarea. Dacă nici o potrivire adecvată nu este găsită în fereastră, sunt trei tehnici principale pentru a face codarea să reuşească. Metoda utilizată în lucrarea originală a lui Ziv şi Lempel[50] este de a cere ca un caracter să fie întotdeauna transmis după fiecare pointer. O mai bună compresie este realizată prin transmiterea unui caracter numai atunci când este necesar. Aceasta necesită transmiterea unui bit suplimentar la fiecare pas de codare, pentru a indica dacă următorul cod reprezintă un pointer sau un caracter literal. Storer şi Szymanski[52] şi Bell[53] dau exemple de aplicare a acestei tehnici. O a treia metodă profită de observaţia că, în general, caracterele noi tind să apară în grupuri. Într-o astfel de metodă se transmite un cod pentru a indica numărul caracterelor fără potrivire, care apoi sunt transmise consecutiv.[54]

Majoritatea metodelor LZ77 presupun că se utilizează o căutare maximală pentru a realiza împerecherea, cu alte cuvinte, se foloseşte *cea mai lungă* potrivire pentru următoarele câteva caractere. Acest lucru nu este în mod necesar optim, întrucât a lua o potrivire lungă poate împiedica o altă bună împerechere care urmează. Unele sisteme vor evalua beneficiul codării explicite a caracterului următor (sau caracterelor), pentru a vedea dacă o potrivire mai lungă este găsită pornind de la o poziţie mai târzie a intrării. Dacă se merge prea departe, aceasta poate încetini considerabil codorul, dar majoritatea îmbunătăţirilor compresiei pot fi atinse privind înainte doar un pas.[55]

Structuri de căutare rapidă a datelor pot fi aplicate fazei de codare, şi, în mod particular, performanţe bune pot fi atinse prin combinarea lor. Combinaţia dintre decodarea foarte rapidă, codarea rapidă şi performanţe relativ bune de compresie, au făcut tehnicile LZ77 foarte populare ca metode de compresie de folosinţă generală şi ele îşi câştigă o largă popularitate şi prin faptul că funcţionează eficient chiar şi pe cele mai modeste calculatoare personale. LZ77 este utilizată de programele de compresie de folosinţă generală, de software-ul de arhivare de fişiere şi de sistemele de compresie *„on-the -fly"* pentru discuri. O variantă mai rapidă de LZ77 este tehnica LZRW1.[56]

Familia de metode LZ78 utilizează o tehnică diferită pentru construirea dicţionarului. Ele construiesc un dicţionar explicit prin spargerea intrării în fraze şi înmagazinarea lor într-o structură de date pentru potrivire rapidă. În mod uzual, frazele sunt fără suprapunere, astfel încât sunt disponibile pentru codare mai puţine coloane de dicţionar decât pentru metodele LZ77. Deşi aceasta înseamnă că sunt posibile mai puţine substituţii, codurile vor fi mai scurte. În general vorbind, metodele bazate pe LZ78 tind să ofere performanţe de compresie mai slabe decit cele corespunzătoare bazate pe LZ77, iar decodarea este şi ea mai scumpă deoarece decodorul trebuie să-şi construiască propriul său dicţionar. Oricum, codarea are tendinţa de a fi mai rapidă întrucât este necesară mai *puţină muncă pentru întreţinerea şi căutarea în dicţionar.*

Metoda originală LZ78 construită de Ziv si Lempel[51] este ilustrată de exemplul din figura 5. Caracterele codate anterior sunt descompuse în fraze (numerotate dedesubt) utilizând o euristică simplă şi pasul de codare caută cea mai lungă frază care se potriveşte cu caracterele care urmează să fie codate. Virgulele arată modul în care textul anterior a fost descompus în fraze pentru dicţionar. În acest caz, fraza numărul 127, *abaac*, este cea mai lungă potrivire pentru caracterele care urmează. Este reprezentată prin transmiterea numărului 127, urmat de caracterul imediat următor celor care s-au potrivit, care în acest caz este un *b*. Linia abia transmisă (*abaacb*, în exemplu) este apoi adaugată în dicţionar. Cu alte cuvinte, algoritmul de analiză este pur şi simplu de a adăuga un caracter la o frază existentă din dicţionar. Algoritmul poate fi implementat în mod natural, utilizând o structură de date arborescentă (*trie*).[2]

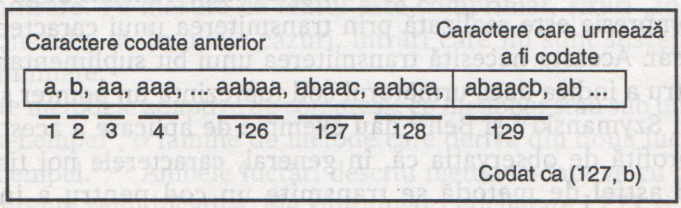


Figura 5. Un pas de codare pentru metoda de compresie LZ2

Ca și în cazul LZ77, există mai multe variante ale metodei originale LZ78, utilizând diferite euristici de analiză și diferite metode pentru reprezentarea zonelor de potrivire. Câteva din cele mai larg răspândite variante de LZ78 sunt bazate pe o metoda propusă de Welch[57], consacrată sub denumirea LZW. Metoda LZW elimină caracterul explicit din perechea LZ78 (cod, caracter). Aceasta se face prin inițializarea dicționarului pentru a conține toate caracterele din alfabet, ceea ce garantează că o potrivire va fi întotdeauna găsită – chiar și dacă numai un singur caracter este potrivit. Fraze noi sunt create pentru dicționar, prin extinderea unei fraze codate adăugându-i primul caracter din fraza următoare.

Acest lucru este ilustrat în figura 6, unde virgulele arată grupurile de caractere care au fost codate ca fraze de sine stătătoare. Sub fiecare frază este arătată cea nouă, construită din ea, care este obținută prin adăugarea primului caracter din fraza următoare. Cea mai lungă frază care se potrivește cu caracterele în curs de codare este numărul 109, respectiv *aabaa*. Acesta este transmis și noua frază, *aabaaa*, este adăugată în dicționar ca numărul 112. O capcană a acestei abordări este aceea că decodorul nu poate construi o frază decât după folosirea ei, deoarece nu va avea primul caracter al frazei următoare. De exemplu, dacă pasul următor de codare a descoperit că fraza cu numărul 112 a fost cea mai lungă potrivire, decodorul nu va avea fraza disponibilă. Din fericire, există o soluție simplă a acestei probleme, întrucât decodorul nu are nevoie să știe întreaga frază 113 pentru a fi capabil să construiască fraza 112, ci numai de primul caracter. Acesta este deja cunoscut – trebuie să fie același cu primul caracter al frazei 112 și astfel decodarea poate decurge în pofida aparenței de referință circulară.

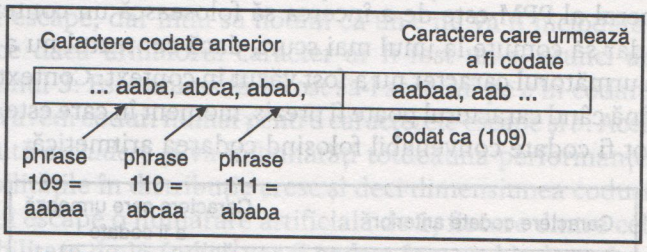


Figura 6. Un pas de codare pentru metoda de compresie LZW

3.3. *Echivalența modelelor*

Deși am făcut distincția dintre metodele bazate pe context și metodele tip dicționar, linia de departajare dintre cele două este estompată. De exemplu, este posibil să utilizezi contextul pentru a prezice intrările de dicționar[58,59] și este posibil pentru sistemele bazate pe context să utilizeze șiruri ca simbolurile lor (precum codoarele bazate pe cuvinte, menționate anterior).

Mai mult decât atât, un efort considerabil a fost depus pentru a examina dualitatea dintre cele două abordări, folosind tehnica blocurilor (combinarea codării câtorva caractere consecutive) și descompunerea (spargerea codării unui simbol în câțiva pași). O metodă bazată pe context poate fi convertită într-o metodă dicționar cu ajutorul blocurilor. Pe măsură ce blocurile devin mai mari, dependențele între blocuri devin mai puțin semnificative și metoda dicționar devine o bună aproximare

a celei bazate pe context. Oricum, dicționarul va deveni imens și nu va mai fi practic pentru implementare.

Invers, s-au dat câteva tehnici pentru descompunerea frazelor din metoda dicționar în coduri de simboluri discrete. Lungimile codurilor pentru simboluri individuale dintr-o frază sunt alese astfel încât sumate să dea aceeași lungime ca și fraza din metoda dicționar. Descompunerea a fost găsită pentru metoda originală LZ78,[2] LZ77,[60] și pentru unele clase mai generale.[1,3] Din nou, această transformare nu conduce la metode practice, dar demonstrează o relație strânsă între cele două metode. Uzual, o descompunere expune metoda dicționar la ineficiențe, ceea ce ajută la a explica de ce metodele bazate pe context dau de obicei rezultate de compresie mai bune.

În general metodele dicționar sunt mai rapide deoarece ele procesează câteva caractere o dată, în timp ce metodele bazate pe context dau compresie mai bună deoarece folosesc bine informațiile contextului. Următoarele capitole descriu unele metode practice, bazate pe tehnicile de modelare descrise mai sus.

3.4. Algoritmul PPM

PPM este abrevierea pentru *Prediction by Partial Matching* (= predicție prin potrivire parțială), o metodă propusă prima dată în 1984.[61] Este o metodă adaptivă bazată pe modele contextuale, care asigură performanțe de compresie extrem de bune pe o serie de tipuri de fișiere, în particular pe fișierele text. Pe fișierele de text în engleză, atinge rate de compresie tipice de puțin peste doi biți per caracter. Principalul ei neajuns este de a fi considerabil mai lentă decât metodele rapide Ziv-Lempel.

Principiul general al PPM este de a încerca să folosească un context lung pentru a face o predicție, dar să comute la unul mai scurt, dacă contextul nu a mai fost întâlnit înainte, sau dacă următorul caracter nu a fost văzut în context. Contextul este trunchiat în mod repetat până când caracterul poate fi prezis, moment în care este codat. Predicțiile făcute de PPM pot fi codate convenabil folosind codarea aritmetică.

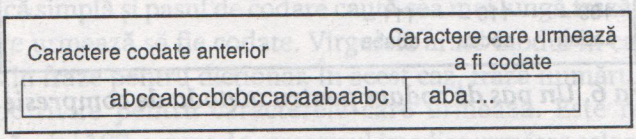


Figura 7. Un pas de codare PPM

Figura 7 arată ce se întâmplă pentru a coda un caracter folosind PPM. Sistemul este pe cale de a coda un caracter *a* și contextul este *...abaabc*. PPM începe cu un context de lungime maximă, tipic în jur de patru caractere. În cazul din exemplu, aceasta ar da contextul de ordin 4 *aabc*. Codorul caută o structură de date construită din textul codat anterior, pentru apariții anterioare ale contextului. Dacă nici una nu este găsită, codorul comută automat către un context de ordinul 3. Decodorul poate face la fel, deoarece el are la dispoziție exact același text. Eventual, atât codorul cât și decodorul se vor stabili pe contextul maxim care a apărut anterior, fără a fi necesară nici o comunicare între cele două. Tabelul 4 arată de câte ori a fost găsit fiecare context pentru exemplul din figu-ra 7. Contextul de ordinul 3, *abc*, este cel mai larg care a fost găsit înainte și el va fi ales inițial pentru a coda caracterul următor. Atât codorul cât și decodorul mențin o socoteală

a caracterelor care au apărut în acest context. În exemplu a apărut, numai caracterul *c*, de două ori.

Tabelul 4. Un tabel de contexte pentru pasul de codare din figura 7

Context		Numărătorul de caractere	Numărul total
Ordinul 4	*aabc*	0	
Ordinul 3	*abc*	2	*c*(2)
Ordinul 2	*bc*	4	*c*(3), *b*(1)
Ordinul 1	*c*	8	*c*(3), *b*(2), *a*(3)
Ordinul 0		22	*c*(9), *b*(6), *a*(7)

Aceasta ar părea să ceară ca *c* să fie prezis cu o probabilitate de 100%, dar aceasta va opri orice alt caracter să fie codat și, în exemplu, ar face imposibilă codarea următorului *a*, deoarece el nu a mai fost văzut niciodată în acest context înainte. Pentru a debloca această situație, se adaugă în context un caracter adițional, noțional, numit caracter *„escape"* (scăpare). Codorul transmite un semnal „escape" pentru a indica decodorului că acest context nu poate fi folosit și amândouă comută în jos la contextul următor mai mic. În același timp, vom da caracterului escape o numărare de 1, mărind artificial numărul de situații în care a fost văzut contextul de ordin 3 și dând caracterului escape o probabilitate de 1/3. Dacă s-ar fi codat un *c*, el ar fi fost reprezentat folosind cele 2/3 din spațiul de probabilitate rămas.

În această etapă a exemplului, atât codorul cât și decodorul folosesc acum contextul de ordinul 2. Acest context poate fi folosit pentru a prezice un *c* sau un *b*, dar nu un *a*. Este cerut un alt escape, dar întâi să notăm că un *c* nu va fi niciodată codat în această situație, deoarece dacă următorul caracter ar fi fost un *c*, atunci ar fi fost codat în contextul de ordinul 3. Este atunci fără rost să-l socotim pe *c* în codarea de ordinul 2 și astfel ajungem să avem coduri numai pentru caracterele escape și *b*. Această îmbunătățire simplă este numită *excludere* și va îmbunătăți totdeauna performanțele de compresie, deoarece probabilitățile în distribuție cresc și deci dimensiunea codurilor scade. Dacă i se dă caracterului escape o numărare artificială de 1, fiecare dintre cele două caractere va avea o probabilitate de ½ (adică vor fi codate într-un bit de codorul aritmetic).

Odată ce codorul a ajuns la contextul de ordinul 1, caracterul *a* poate fi codat deoarece el are o numărătoare de valoare diferită de zero. În acest context, *b* și *c* vor fi excluse și singurele caractere în distribuția de probabilități vor fi *a* și escape. Caracterul escape este necesar deoarece codorul are nevoie să ofere posibilitatea apariției unui alt caracter. Dacă numărătorul pentru escape este unu, atunci *a* va fi codat cu o probabilitate de ¾.

Dacă ar trebui codat un caracter nou, să zicem *d*, PPM va adăuga escape până jos, la un context de ordinul 0, apoi va scăpa de acel context (tot cu escape) și va coda noul caracter folosind o distribuție de probabilitate uniformă peste toate caracterele posibile. Aceasta va apărea frecvent în apropierea începutului textului, dar scăparea (caracterul escape) va fi folosită din ce în ce mai rar pe măsură ce a fost văzut suficient text pentru a construi un model bun.

O problemă importantă de proiectare este decizia asupra modului de calcul al probabilităților caracterului escape. Aceasta implică estimarea probabilității ca un element nemaivăzut înainte să apară, ceea ce se numește *problema frecvenței-zero*.[62] S-au propus

câteva abordări, inclusiv cea simplă, menționată mai sus, de a aloca un numărător de valoare unu caracterului escape. O metodă ce pare să funcționeze consistent bine în practică se bazează pe numărul de caractere distincte văzute deja în contextul curent. Dacă s-au văzut r caractere distincte într-un total de n apariții ale contextului, atunci probabilitatea unui escape este estimată la $r / (n + r)$ și probabilitatea rămasă de $n / (n + r)$ este distribuită proporțional între celelalte caractere. Această abordare a fost consacrată ca „metoda C" și varianta de PPM care o folosește se numește PPMC. O descriere completă a PPMC, inclusiv a structurii de date pentru implementarea ei eficientă, este dată de Moffat.[63] Metoda PPMC dă tipic o compresie de puțin peste doi biți pe caracter pentru text în limba engleză. O metodă următoare „D" a fost descrisă[64] și conduce la o compresie ceva mai bună decât metoda C.

Cleary *et al.*[65] descriu o variantă a PPM, numită PPM*, care lucrează cu contexte de lungimi arbitrare. Se folosește o euristică pentru a determina lungimea contextului pentru fiecare predicție. Burrow și Wheeler[66] au dezvoltat un alt algoritm, care la prima vedere pare a fi destul de diferit, dar care a fost arătat de Cleary *et al.* ca fiind un echivalent semi-static al PPM*. Metoda Burrow și Wheeler are, însă, avantajul de a fi mult mai rapidă în practică.

3.5. *Algoritmul ZIP*

ZIP este un algoritm folosit într-un număr de programe de compresie freeware (necondiționată) făcute pentru prima dată disponibile de către grupul *Info-Zip*.[b] Programele care îl folosesc includ *zip*, *gzip*, *pgp* și majoritatea implementărilor PNG. Sistemul *pkzip* folosește același format de fișier și, prin urmare, algoritmi similari de codare și decodare.

Metoda ZIP este o variantă înalt optimizată a metodei de compresie LZ77 și folosește resurse de procesare relativ modeste pentru a obține o foarte bună compresie. Stabilirea compromisului între compresie și viteză este accesibil utilizatorului, făcându-l foarte util într-o varietate de situații.

ZIP folosește un tabel combinat (*hash table*) pentru a localiza aparițiile precedente de șiruri. Următoarele trei caractere care trebuie codate sunt combinate (*hashed*) și valoarea rezultată este folosită pentru a căuta o poziție de intrare în tabel. Aceasta este capul unei liste corespondente, care conține locurile unde cele trei caractere au apărut în fereastră. Lungimea listei corespondente este limitată astfel încât să evite creșterea prea mare a timpului de căutare în listă, cu prețul unei mici pierderi în compresie. Cum lungimea listei se poate alege în momentul codării, utilizatorul poate stabili un raport între viteză și compresie. A avea o limită este în mod particular important atunci când intrarea conține șiruri lungi ale aceluiași caracter, deoarece acestea vor produce o listă foarte lungă de referințe. Listele lungi consumă mult timp pentru a fi întreținute și sunt în general ne-necesare deoarece primele câteva elemente din listă vor fi cel mai adesea suficiente pentru a găsi o potrivire bună. Compresia poate fi îmbunătățită puțin prin stocarea aparițiilor recente la începutul listei corespondente, pentru a favoriza potrivirile recente. Potrivirile sunt reprezentate în fișierele codate printr-un pointer constând dintr-un offset și o lungime. Dacă nu poate

[b] `http://quest.jpl.nasa.gov/Info-ZIP`

fi găsită nici o apariție anterioară potrivită, se transmite un caracter „brut" (nou, neutilizat încă). Offset-ul unui pointer este reprezentat utilizând un cod de minimă redundanță, astfel încât cele mai frecvente offset-uri (în general cele mai recente) sunt codate pe mai puțini biți. Lungimea de potrivire este reprezentată printr-un alt cod prefix; și acest cod este utilizat atunci când sunt de reprezentat caractere brute (noi). Ar putea să apară contradictorie combinarea lungimilor de potrivire și a caracterelor brute într-un singur cod, dar aceasta poate da o mai bună compresie decât separarea lor, întrucât în acest din urmă caz ar fi trebuit transmis un bit suplimentar pentru a indica dacă următoarea intrare este o lungime de potrivire sau un caracter, în timp ce codul combinat poate utiliza mai puțin decât un bit, în medie, pentru a transmite aceeași unitate de informație. Lungimea de potrivire este trimisă înaintea offset-ului unui pointer, astfel încât decodorul să poată spune dacă este în curs de transmitere un pointer sau un caracter brut.

Dacă utilizatorul specifică faptul că se consideră compresia mai importantă decât viteza, ZIP utilizează o formă simplă de analiză non-maximală pentru a îmbunătăți performanțele compresiei, prin transmiterea unui caracter brut chiar dacă poate fi folosit un pointer. Această situație apare când utilizarea unui caracter brut oferă o mai bună potrivire pentru caracterele imediat următoare celui care este în curs de codare.

Codurile pentru ZIP sunt generate semi-static. Blocuri de până la 64 KB sunt procesate o dată. Codurile canonice adecvate sunt generate pentru pointere și caractere brute și un tabel de coduri este asociat provizoriu formei comprimate a blocului. Aceasta înseamnă că ZIP nu este cu adevărat o metoda cu trecere unică. Oricum, blocurile sunt suficient de mici pentru a fi ținute în memorie. În consecință, fișierul de intrare are nevoie să fie citit numai odată și programul se comportă ca și cum ar opera într-un singur pas.

Datorită algoritmului de căutare rapidă și reprezentărilor compacte de ieșire, bazate pe coduri de minimă redundanță, ZIP oferă performanțe de ieșire mai evoluate decât alte metode Ziv-Lempel, atât în ceea ce privește viteza cât și compresia efectivă. Comprimă în mod tipic fișierele de text în limba engleză la sub 3 biți pe caracter. Aceasta este remarcabil de aproape de performanța celor mai bune modele bazate pe context și încă ZIP funcționează foarte repede.

3.6. *Algoritmul COMPRESS*

Utilitatea COMPRESS a Unix-ului este una dintre cele mai larg utilizate variante ale metodei LZW. A început ca o implementare a LZW, dar a fost fin adaptată pentru a o face potrivită ca sistem de compresie de utilizare generală. Popularitatea sa mare dovedește performanțele sale, deși recent a fost înlocuită de ZIP, care poate obține o mai bună compresie și performanțe de viteză în aceleași circumstanțe, în mod special dacă a decoda continuu este important.

În COMPRESS, numărul de biți utilizați pentru a reprezenta fraze este gradual crescut în timpul codării, astfel încât sunt disponibile doar atâtea coduri câte sunt suficiente pentru a reprezenta numărul de intrări în dicționar. De asemenea, COMPRESS plasează un maxim al numărului de fraze care sunt construite, ceea ce limitează cantitatea de memorie cerută pentru codare. Când dicționarul este complet, adaptarea încetează. Performanța compresiei este monitorizată și, dacă se deteriorează semnificativ,

dicționarul este șters și apoi reconstruit de la început. Dicționarul este înmagazinat într-o structură arborescentă de date (*trie*) și accesul la structura de date este foarte rapid, utilizând o adresare combinată. Ramificația unui nod în structură este găsită combinând numărul nodului și eticheta ramurii care este urmărită.

Această mixare dintre combinare și căutarea fină a codurilor face COMPRESS relativ rapid și oferă o bună compresie. În mod tipic, COMPRESS reduce fișierele de text în engleză la aproximativ 3,5 biți pe caracter. Aceasta este mult sub jumătatea mărimii lor originale, dar mai rămâne un drum lung până la performanțele unor metode ca PPM. Oricum, acum este înlocuit de ZIP care, având parametrii corecți, poate da performanțe similare, dar poate fi utilizat, de asemenea, pentru a da o mai bună compresie sau o mai mare viteză, modificând întinderea căutării sale.

3.7. *Performanța relativă*

Testul final al gradului de utilitate a programului de compresie este performanța sa. Figura 8 arată comportarea câtorva dintre metodele descrise aici, pe o colecție de texte în engleză de 2 GB (gigaocteți). Figura schițează viteza de decodare (în megaocteți pe minut, pe un Sun Microsystems SPARC 10 Model 512), în relație cu rata de compresie atinsă (măsurată ca procentul rămas după compresie). În multe cazuri, viteza de codare este similară cu viteza de decodare. Opțiunile ZIP au fost stabilite mai curând pentru viteză decât pentru eficacitatea compresiei și o compresie ușor mai bună este posibilă dacă se face o setare alternativă. Metoda HUFFWORD este un model de ordinul zero, bazat pe cuvânt și utilizând codarea canonică semi-statică[13].

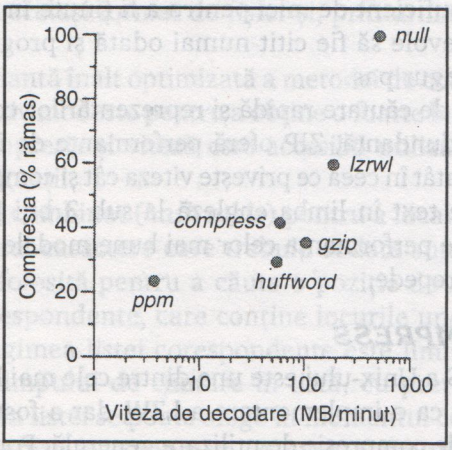


Figura 8. Performanța compresiei și rata de decodare

Este un compromis clar între viteza de ieșire și eficiența compresiei. Cea mai bună metodă este PPM, dar este de asemenea cea mai lentă. Cea mai rapidă metodă este *nulul*, cu alte cuvinte fără nici o compresie și o copie nealterată a fișierului. Cea mai bună din metodele LZ78 – COMPRESS – este depășită de ZIP atât din punctul de vedere al eficienței compresiei cât și din cel al vitezei. Pentru a prezenta interes, metodele viitoare trebuie să se așeze în afara curbei de înfășurare „dreapta-jos" din figura 8, sau să aibă alte atribute specifice care să recomande utilizarea lor.

3.8. *Viitorul compresiei de text*

În cursul fiecărui an cercetătorii descoperă metode de compresie a textului ceva mai evoluate, deşi aceasta se obţine în principal prin intermediul căutării fine – marile acumulări au fost făcute la sfârşitul anilor 1970 (codarea Ziv – Lempel şi codarea aritmetică) şi începutul anilor '80 (modele contextuale, ca de exemplu PPM). Apare ca fiind în mod particular dificilă depăşirea barierei de compresie în mai puţin de 2 biţi per caracter pentru un text „tipic" în engleză.

Anumite experimente timpurii făcute de Shannon[67], Cover şi King[68] au încercat să găsească o graniţă mai joasă pentru compresibilitatea textului în engleză, având subiecţi umani care să facă predicţiile pentru un sistem de compresie. Aceşti autori au concluzionat că limba engleză are o entropie inerentă de aproximativ 1 bit per caracter, şi că este în general puţin probabil să fim capabili să realizăm o compresie a acestei limbi la o rată mai bună decât aceasta.

Au fost propuse modele bazate pe maşini cu număr finit de stări,[69,70] dar în prezent acestea se dovedesc a nu fi mai puternice decât modelele bazate pe context.[71,72] Inovaţii mai recente includ metode bazate pe dicţionar, care înmagazinează dicţionare multiple şi aleg care dintre ele să fie folosite în funcţie de contextul curent[59], cât şi metode bazate pe analiza gramaticală ne-maximală.[55] Oricum, aceste metode aduc doar mici îmbunătăţiri. Modele mai performante, bazate pe gramatici şi semantici, au fost curtate, dar sistemele practice sunt încă aşteptate să iasă la suprafaţă în căutarea compresiei la mai puţin de doi biţi pe caracter.

4. Modele pentru imagini

Sunt trei domenii principale în care metodele de compresie fără pierderi sunt utilizate pentru transmisia imaginilor. Primul se referă la imaginile cu două niveluri – notabil faxurile – pentru care au fost dezvoltate câteva standarde. Al doilea este pentru aplicaţiile de specialitate ale imaginilor în scara de gri (sau unele cu o paletă de culori limitată), cum ar fi razele X din medicină, unde informaţia de imagine este subtilă şi consecinţele unei erori atât de groaznice încât utilizatorii nu pot să accepte nici cea mai mică pierdere. Al treilea este pur şi simplu din raţiuni de convenienţă: în practică imaginile sunt frecvent stocate în calculatoare şi transmise între ele în formate comprimate fără pierderi deoarece pur şi simplu nu merită să ne îngrijorăm cu privire la gradul de aproximare care ar putea fi tolerat. Tehnici cu pierderi sunt utilizate pentru majoritatea celorlalte aplicaţii de imagine, întrucât, cum cantitatea de date asociate cu fiecare pixel creşte de la 1 bit pentru imaginile cu două niveluri la câţiva biţi pentru imaginile în scară de gri şi apoi la un număr mai mare de cuvinte pentru imaginile cu adevărat color, îmbunătăţirea în compresie, care rezultă din acceptarea reprezentărilor cu pierderi mici creşte, iar efectul perceptual al unei slabe degradări tinde să scadă.

La sfârşitul anilor '70, standardul internaţional de comunicaţii CCITT a recunoscut potenţialul tehnologiilor care apăreau pentru transmisiile facsimil şi a cerut un standard pentru fax. Acesta a fost finalizat în 1980, ca standardul „Group 3" şi formează bazele aparatelor de fax care sunt acum în uzul de fiecare zi. Standardul încorporează tehnici simple de compresie de date, bazate pe codarea lungimilor sub-şirurilor de biţi succesivi identici (*run-length*), codarea prefixelor şi codarea diferenţială, pentru a exploata

coerența între linii succesive.[73] Puțin după aceea a fost publicat standardul „Group 4", care este în esență același cu Group 3, dar este destinat utilizării în rețele digitale, mai degrabă decât în circuitele telefonice analogice convenționale. Ambele tehnici sunt descrise în capitolul următor.

În anii '80, adoptarea graduală a codării aritmetice a îndepărtat orice constrângere artificială asupra tipului de model care ar putea fi utilizat, determinând cercetătorii să înceapă să investigheze folosirea modelelor contextuale pentru compresia imaginilor cu două niveluri.[74] Așa cum am explicat mai sus, modelele contextuale de text condiționează distribuția de probabilitate a caracterului care urmează, de câteva caractere precedente. Pentru fiecare context diferit care apare, atât codorul cât și decodorul mențin o distribuție de probabilitate a caracterului care urmează. Caracterul real care urmează este codat ținând seamă de această distribuție (folosind codarea aritmetică) și refăcut fără eroare de către decodor, deoarece acesta are acces la exact aceeași distribuție. Aceasta funcționează și pentru imagini cu două niveluri, așa cum s-a descris în capitolele 4.2 și 4.3. Modelarea contextuală formează baza unui recent standard CCITT/ISO pentru compresie fără pierderi a imaginii, numit JBIG, care este prezentat în continuare.

Pentru imagini în tonuri continui (i.e. cu scară de gri sau color), unele metode de compresie cu pierderi încorporează moduri fără pierderi în care diferența între versiunea cu pierderi și cea originală este codată și transmisă într-un anume fel. De exemplu, recentul standard JPEG reproduce imaginile cu tonuri continue cu o calitate excelentă la circa 1 bit per pixel.[75] Un bit per pixel, punctul de plecare pentru fax, este o compresie impresionantă pentru o imagine cu nuanțe de gri sau color, care, în general, are între 8 și 36 de biți per pixel după digitizare. Deși este o metodă cu pierderi, care nu reconstruiește imaginea originală exact, JPEG include o opțiune de compresie fără pierderi, pentru situații în care aceasta se consideră a fi necesară.

Compresia de imagine cu pierderi este în general bazată pe principii destul de diferite de metodele fără pierderi; de exemplu, folosirea analizei spectrale mai degrabă decât predicția bazată pe context. Oricum, recent s-a manifestat o deplasare a interesului către metodele de compresie fără pierderi a imaginii. În capitolul 4.5 descriem popularul format GIF, care este bazat pe codarea Ziv-Lempel a secvenței valorilor de pixeli și este larg folosit pentru compresia imaginii în practică. Metode special proiectate, care iau în considerare contextul bidimensional, dau rezultate mult mai bune și vom descrie în capitolul următor o metodă dezvoltată special pentru compresia fără pierderi a imaginilor cu scară de gri. Deși este mult mai simplă decât JPEG fără pierderi, performanțele sale apar destul de favorabile. În sfârșit, în capitolul 4.7, vom scoate în evidență o metodă mult mai elaborată care a fost introdusă foarte recent și care reprezintă vârful de lance în performanțele de compresie.

4.1. *Standardul CCITT pentru facsimil*

Standardul CCITT începe prin a specifica detalii ca dimensiunea hârtiei și rezoluția scanării. Un document de o pagină la rezoluția standard de 200×100 dpi (*dots per inch* = puncte per inch) conține aproximativ 1700×1200 de pixeli albi-sau-negri, sau 2 Mbiți de informație înaintea compresiei. Rata de transmisie este în mod normal 4800 biți pe secundă, care la momentul când a fost introdus standardul se puteau obține confortabil

pe o linie telefonică obişnuită. Dacă nu s-ar fi folosit nici o compresie, ar fi fost necesare 430 de secunde, sau aproximativ şapte minute, pentru a transmite o pagină de document. Aceasta contrastează cu un timp mediu, pentru documente tipice, de aproximativ un minut pe pagină, folosind standardul Group 3.

Standardul specifică două metode de codare: o schemă unidimensională care tratează independent fiecare linie scanată şi una bidimensională care exploatează coerenţa între linii succesive. De fapt, a doua schemă identifică poziţiile de-a lungul fiecărei linii în care imaginea se schimbă din negru în alb, sau din alb în negru şi le codează corelat cu cu poziţiile corespunzătoare din linia precedentă. Pentru a porni lucrul, schema unidimensională este folosită la transmisia primei linii.

Fiecare linie scanată este privită ca o secvenţă de şiruri alternative de puncte albe şi negre. Liniile sunt presupuse a începe cu un şir alb, astfel că receptorul poate păstra sincronizarea şi liniile care încep cu negru sunt tratate ca şi cum ar începe cu un şir alb de lungime zero. În schema unidimensională, o linie este reprezentată prin codarea lungimii fiecărui şir, folosind un cod prefix, ne-adaptiv, pre-specificat. Tabele de coduri separate sunt folosite pentru şiruri alb şi negru, deoarece distribuţia lor statistică este diferită – imaginile textuale sunt în mod normal negru pe alb, iar şirurile negre de câţiva pixeli sunt mult mai comune decât şirurile albe de aceeaşi lungime. Fiecare tabel de coduri poate reprezenta o lungime de şir de până la 63 de pixeli şi cuvinte speciale de cod „de machiaj" pot controla grupuri de multipli de 64 de pixeli. Tabelele de coduri au fost obţinute prin optimizarea codurilor pentru un set particular de documente text.

Liniile codate se termină printr-un cuvânt de cod special, sfârşit-de-linie. Acesta este ales a fi o secvenţă unică binară care nu poate apărea într-o linie validă de date codate. Chiar dacă apar erori de transmisie care deteriorează datele liniei scanate şi distrug sincronizarea receptorului cu cuvintele de cod, sfârşitul de linie va fi încă detectat cu mare siguranţă.

Codul bidimensional este o schemă linie-cu-linie în care un rând de pixeli, „linia curentă", este codat ţinând seamă de precedentul rând, deja transmis, „linia de referinţă". Acesta este numit „modul vertical". Poziţia fiecărei schimbări de culoare (alb în negru sau negru în alb) în linia curentă este codată în corelaţie cu o schimbare de poziţie apropiată (a aceleiaşi culori) în linia de referinţă, dacă există una. „Apropiată" este înţeles în limitele a trei pixeli şi astfel codarea în modul vertical poate lua una dintre cele şapte valori: -3, -2, -1, 0, +1, +2, +3. Dacă nu este nici o poziţie de schimbare apropiată în linia de referinţă, se foloseşte codul unidimensional obişnuit, numit „modul orizontal". A treia condiţie apare când linia de referinţă conţine un şir care nu are nici un corespondent în linia curentă şi atunci un cod „pass" (de trecere) este transmis pentru a semnaliza receptorului că următorul şir complet, de culoarea opusă, din linia de referinţă, trebuie sărit.

Această schemă de codare bidimensională este în mod special susceptibilă la erori, deoarece orice eroare de transmisie dintr-o linie se va propaga în jos de-a lungul întregii pagini. Prin urmare, unele linii se transmit folosind codarea unidimensională, iar cea bidimensională este folosită numai pentru celelalte, care rămân. Standardul Group 3 face ca cel puţin fiecare a k-a linie să fie transmisă folosind metoda unidimensională şi k este setat la 2, astfel încât o eroare poate distruge cel mult două rânduri consecutive. Standardul include, de asemenea, un mod de înaltă rezoluţie cu dublul densităţii

verticale (200×200 dpi) și pentru acesta k este setat la 4, astfel încât să se poată obține o compresie mai bună. Erorile pot distruge atunci până la 4 linii, dar aceasta corespunde la numai două linii de rezoluție standard.

Îmbunătățirea de compresie a codării bidimensionale Group 3 este limitată de faptul că fiecare a k-a linie este codată unidimensional, pentru a localiza efectul propagării erorilor. În sistemele de transmisie digitale, unde erorile sunt detectate și corectate la un nivel mai mic de protocol, acest lucru nu este necesar: schema de codare bidimensională poate fi folosită pentru fiecare linie din pagină, cu condiția asumării unei linii albe de referință la începutul paginii, pentru a face ca procesul să pornească. Acesta este standardul facsimil Group 4. Prin urmare, singura diferență între Group 3 și 4 este că ultimul dispune de facilități care sunt proiectate pentru a-i asigura robustețe în fața erorilor de transmisie, cu condiția ca acestea să fie corectate la nivele joase de protocol. Deoarece abilitatea de detecție și corecție a erorilor implică redundanțe de reprezentare, omisiunea acestora duce la o creștere și mai mare a compactității codării și, de aici, a vitezei de transmisie. Mai multe informații despre standarde și rezultatele compresiei imaginilor de test CCITT folosind aceste standarde pot fi găsite în lucrările [76], [77] și [78].

4.2. *Modele contextuale*

Un model contextual condiționează distribuția de probabilitate a unui pixel de a fi alb sau negru, de care dintre pixelii precedenți au fost albi și care dintre aceștia au fost negri. În timp ce în cazul codării textului este clar că cele mai bune litere (de referință), pe baza cărora putem să condiționăm distribuția, sunt cele precedente; în cazul imaginii trebuie să decidem dacă să folosim pixelii de deasupra celui curent, sau de la stânga, sau, cel mai bine, și pe unii și pe ceilalți.

Cel mai bun plan este de a folosi pentru context un șablon cu pixelii care înconjoară pe cel curent, dar care îl preced în ordinea de transmisie. Experimentele au arătat că se obțin rezultate bune cu șablonul din figura 9. Un punct negru marchează fiecare pixel inclus în șablon și un cerc magic (sub formă de țintă – *bull's eye*) marchează poziția pixelului care urmează a fi codat. Pixelii gri sunt cei ale căror valori nu sunt cunoscute de decodor și deci nu pot fi incluși în context. Cercurile albe sunt pixeli ale căror valori sunt cunoscute dar care nu sunt incluși în context.

De exemplu, dacă sunt șapte pixeli în șablon, sunt cel mult 128 de contexte diferite. Numărul de apariții ale unui pixel alb și ale unui pixel negru în fiecare din contexte sunt și ele numărate, rezultând cu totul 256 de „numărători" (variante). Atunci, în timpul codării, pixelii din șablon sunt folosiți ca și context pentru a determina care numărătoare să fie folosită. Aceasta se poate implementa prin concatenarea celor șapte biți pentru a forma un număr de context cuprins între 0 și 127, care este folosit pentru a indexa lista numărătorilor de apariții. Numărătorile sunt folosite pentru a comanda un codor aritmetic, așa cum a fost descris în subcapitolul 2.6. De notat că aceasta este una dintre aplicațiile unde codoarele aritmetice sunt cu adevărat necesare – nici un cod fără prefix nu poate reprezenta simbolurile unui alfabet binar în mai puțin de un bit per simbol și împărțirea în blocuri care să constituie șiruri uniforme, care astfel să mărească alfabetul, nu este eficientă deoarece probabilitatea unui 1 este diferită în fiecare context.

Șaisprezece biți sunt suficienți pentru stocarea numărătorilor, pentru că nu se câștigă nimic în plus făcându-le mai precise, și toate numărătorile pot fi înjumătățite ori de

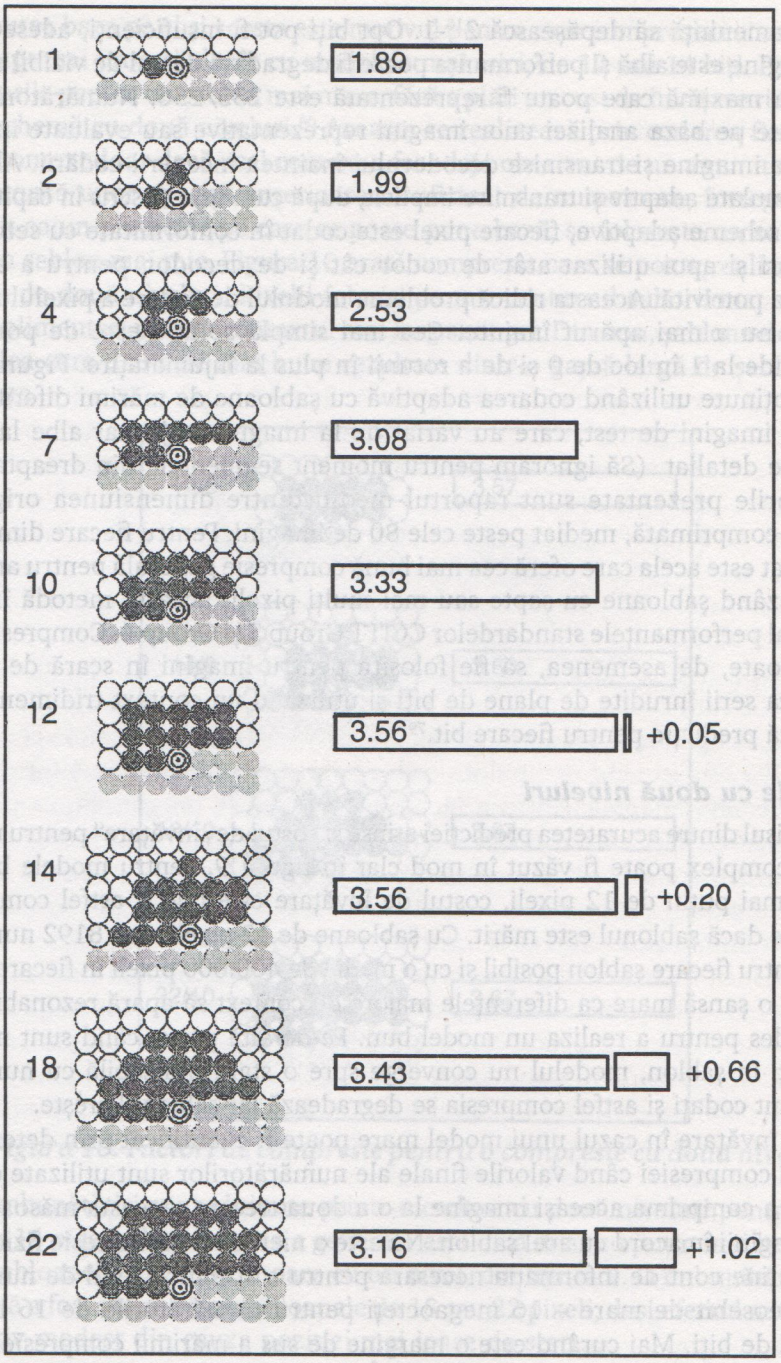


Figura 9. Factori de compresie pentru compresie bazată pe context

câte ori una amenință să depășească 2^{16}-1. Opt biți pot fi insuficienți, adeseori mare parte din imagine este albă și performanța poate fi degradată destul de vizibil deoarece probabilitatea maximă care poate fi reprezentată este 255/256. Numărătorile pot fi predeterminate pe baza analizei unor imagini reprezentative sau evaluate individual pentru fiecare imagine și transmise decodorului înaintea începerii codării. Alternativ, ele pot fi acumulate adaptiv și transmise implicit, după cum este descris în capitolul 2.1. Pe baza unei scheme adaptive, fiecare pixel este codat în conformitate cu setul curent de numărători și apoi utilizat atât de codor cât și de decodor, pentru a actualiza numărătoarea potrivită. Aceasta ridică problema modului de codare a pixelului într-un context care nu a mai apărut înainte. Cea mai simplă soluție este de porni toate numărătorile de la 1 în loc de 0 și de a rotunji în plus la înjumătățire. Figura 9 arată rezultatele obținute utilizând codarea adaptivă cu șabloane de mărimi diferite, pe un set de 80 de imagini de test, care au variat de la imagini majoritar albe la imagini scanate foarte detaliat. (Să ignorăm pentru moment segmentele din dreapta barelor de jos.) Valorile prezentate sunt raportul mediu dintre dimensiunea originală și dimensiunea comprimată, mediat peste cele 80 de imagini. Pentru fiecare dimensiune, șablonul arătat este acela care oferă cea mai bună compresie generală pentru ansamblul testului. Utilizând șabloane cu șapte sau mai mulți pixeli, această metodă întrece la modul general performanțele standardelor CCITT Group 3 și Group 4. Compresia bazată pe context poate, de asemenea, să fie folosită pentru imagini în scară de gri, prin tratarea lor ca serii înrudite de plane de biți și utilizând un context tridimensional în care să se facă predicție pentru fiecare bit.[79]

4.3. *Modele cu două niveluri*

Compromisul dintre acuratețea predicției atinse și costul de „învățare" pentru utilizarea unui model complex poate fi văzut în mod clar în figura 9. Pentru modele bazate pe șabloane cu mai puțin de 12 pixeli, costul de învățare este mic și astfel compresia se îmbunătățește dacă șablonul este mărit. Cu șabloane de 12 pixeli sunt 8192 numărători, câte două pentru fiecare șablon posibil și cu o medie de 400.000 pixeli în fiecare imagine de test există o șansă mare ca diferențele majore de context să apară rezonabil de des, suficient de des pentru a realiza un model bun. Pe de altă parte, când sunt mai mulți pixeli utilizați în șablon, modelul nu converge spre o stare utilizabilă cu numărul de pixeli care sunt codați și astfel compresia se degradează iar șablonul crește.

Costul de învățare în cazul unui model mare poate fi cuantificat prin determinarea îmbunătățirii compresiei când valorile finale ale numărătorilor sunt utilizate ca model static pentru a comprima aceeași imagine la o a doua trecere. Aceasta măsoară „auto-entropia„ imaginii în acord cu acel șablon. Nu este o metodă de compresie fizic posibilă deoarece nu ține cont de informația necesară pentru a specifica setul de numărători, informație deosebit de mare – 16 megaocteți pentru numărătorile de 16 biți și un șablon de 22 de biți. Mai curând este o margine de sus a mărimii compresiei, care se poate obține utilizând acest șablon particular. Îmbunătățirea incrementală este arătată prin barele din partea dreaptă a figurii 9, pentru șabloanele cu 12 sau mai mulți biți. Este minusculă pentru șabloanele mai mici.

Șabloanele cu 18 sau 22 de biți oferă un context foarte detaliat în care următorul bit poate fi prezis, dar fiecare din aceste contexte largi trebuie să fie inițializat înainte

de a ne putea baza pe el și acesta este motivul pentru care compresia cu aceste șabloane este de fapt mai proastă decât cu modelele mai mici, de 12 sau 14 biți. Pentru a învăța numărătorile pentru un șablon mai mare, fără a plăti un cost de învățare ridicat, se poate folosi o schemă cu două niveluri.[80] Aceasta se realizează prin codarea fiecărui pixel în întregul context doar dacă acel context a fost deja observat de un anumit număr de ori. Dacă, datorită unor apariții anterioare insuficient de numeroase, întregul context nu este privit ca un predictor pe care se poate pune bază, se folosește un subset pentru a genera un șablon mai mic. Figura 10 arată compresia care se poate realiza cu șabloane adecvate, de două niveluri. Pixelii folosiți pentru contextul subordonat sunt negri iar pixelii suplimentari pentru contextul mai larg sunt gri. Din nou, șabloanele arătate sunt doar acelea care dau cele mai bune rezultate dintr-o gamă largă de șabloane care au fost testate.

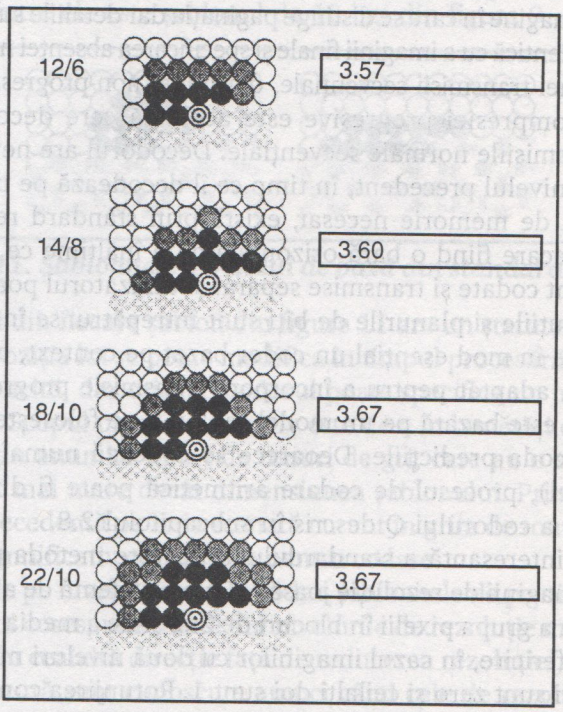


Figura 10. Factori de compresie pentru o compresie cu două niveluri

Mici îmbunătățiri sunt obținute printr-o codare cu două niveluri pentru șabloanele cu 12 sau 14 pixeli, dar compresia generală este încă doar cu puțin mai bună decât cea dată de șabloanele corespunzătoare cu un nivel, din figura 9. O îmbunătățire mult mai accentuată a fost obținută cu șabloanele de 18 sau 22 pixeli, deși efectul net al câștigului este relativ modest din cauza poziției mai joase de start.

4.4. *JBIG: un standard pentru imagini cu două niveluri*

JBIG este un standard recent pentru compresia fără pierderi a imaginilor.[96] Acronimul provine de la *„Joint Bilevel Image Experts Group"*. Este vorba despre un „Joint ... Group" deoarece a fost produs împreună de organismele internaționale de standardizare CCITT

(numit acum ITU) și ISO. După cum sugerează „*Bilevel Image*" (imagini bi-nivel), standardul este proiectat pentru imagini cu două niveluri, deși poate fi folosit și pentru imagini în scară de gri cu un număr mic de biți per pixel, prin comprimarea separată a fiecărui plan de biți.[c] O dată ce rezoluția depășește circa șase biți per pixel, oricum, se poate obține o compresie mai eficientă folosind alți algoritmi.

Metoda de compresie poate fi aplicată întregii imagini sau poate fi folosită într-un mod „progresiv", dând întâi o versiune aproximativă a imaginii, a cărei fidelitate este îmbunătățită în timp. Aceasta se obține pornind cu o imagine cu rezoluție mică și apoi dublând succesiv rezoluția. Imaginea de început este numită nivelul de „bază" și celelalte sunt numite nivelurile „diferențiale". Utilizatorul poate alege rezoluția bazei și numărul de niveluri diferențiale care apar înainte de atingerea nivelului final. Se anticipează că rezoluția cea mai proastă pentru nivelul de bază, folosită în practică, va fi de 12,5 sau 25 dpi, ceea ce dă o imagine în care se distinge paginația dar detaliile sunt ilizibile. Stabilirea rezoluției bazei este identică cu a imaginii finale și specificarea absenței nivelurilor diferențiale duce la realizarea unei transmisii secvențiale, obișnuite, non-progresive.

Un neajuns al compresiei progresive este acela că cere decodorului mai multă memorie decât transmisiile normale secvențiale. Decodorul are nevoie de un buffer în care să poată stoca nivelul precedent, în timp ce îl decodează pe următorul. În scopul reducerii volumului de memorie necesar, există prin standard rezerva de a împărți imaginea în fâșii, fiecare fiind o bară orizontală cu o înălțime ce poate fi definită de utilizator. Fâșiile sunt codate și transmise separat și utilizatorul poate specifica ordinea în care fâșiile, rezoluțiile și planurile de biți sunt întrepătrunse în datele codate.

Metoda JBIG este în mod esențial un codor bazat pe context, cum a fost descris în capitolul precedent, adaptat pentru a încorpora transmisie progresivă. La nivelul cel mai de jos, codarea este bazată pe un model șablon și se folosește codarea aritmetică adaptivă pentru a coda predicțiile. Deoarece se codează numai cantități de un bit (valoarea unui pixel), procesul de codare aritmetică poate fi dinamizat: de fapt se folosește o variantă a codorului Q descris în subcapitolul 2.8.

O caracteristică interesantă a standardului JBIG este metoda pe care o sugerează pentru calcularea imaginii de rezoluție joasă.[81] Calea evidentă de a înjumătăți rezoluția unei imagini este de a grupa pixelii în blocuri de 2×2 și de a media cele patru valori din fiecare bloc. Din nefericire, în cazul imaginilor cu două niveluri nu este clar ce trebuie făcut când doi pixeli sunt zero și ceilalți doi sunt 1. Rotunjirea constantă în plus sau în minus tinde să șteargă imaginea foarte rapid. O altă posibilitate este de a rotunji valoarea într-o direcție aleatoare de fiecare dată, dar aceasta aduce un volum considerabil de zgomot imaginii, mai ales la rezoluțiile cele mai mici. JBIG încorporează o metodă sofisticată de reducere a rezoluției, care păstrează valorile generale de gri. Ea păstrează de asemenea liniile și pixelii izolați de tipul celor care se produc când se convertește o imagine cu trepte de gri într-o imagine cu două niveluri, prin procesul standard de reprezentare prin puncte (ca în pozele din ziar).

---

[c] O performanță mai bună este obținută prin codarea valorii pixelilor utilizând coduri Gray înainte de a diviza imaginea în planuri individuale de biți. Aceste coduri se bucură de proprietatea că valori numerice consecutive sunt reprezentate prin coduri care diferă doar prin poziția unui bit.

Când rezoluția imaginii este redusă printr-un algoritm particular de reducere, se întâmplă uneori ca receptorul să poată determina cu certitudine valoarea unui pixel din pixelii care au fost recepționați anterior. Când se întâmplă așa ceva, pixelul este numit „determinist predictibil". Se poate defini un algoritm care să izoleze acești pixeli și să le aloce valorile lor fără a fi nevoie să se transmită ceva. Aceasta îmbunătățește compresia cu cca. 7%.

Standardul JBIG definește două tipuri de context, unul pentru bază sau pentru nivelul cel mai de jos de rezoluție și altul pentru nivelurile diferențiale sau de rezoluție mai mare. Figura 11 arată șabloanele care pot fi folosite pentru codarea nivelului de bază, care va avea în aplicațiile practice, așa cum s-a arătat la începutul capitolului, fie 12,5, fie 25 dpi. Există două opțiuni care aleg contextul din una sau din două linii precedente de pixeli și ambele implică 10 pixeli. Codorul poate specifica pe care îl folosește. Nu întâmplător, unul dintre ele este șablonul de 10 biți din figura 9.

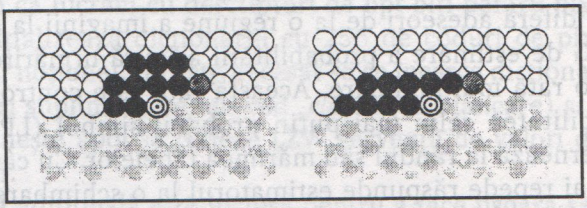


Figura 11. Șabloanele nivelului de bază din standardul JBIG

Singurul pixel gri din fiecare șablon din figura 11 are un statut special. Cunoscut ca un pixel „adaptiv", poziția lui se poate modifica în timpul procesării imaginii. Figura 11 arată poziția lui implicită, dar oricând codorul poate specifica un pixel diferit pentru a fi folosit în locul lui. Scopul pixelului adaptiv este de a îmbunătăți eficiența compresiei în imagini care conțin anumite tipuri de tonuri de gri, care au o structură regulată și care este uzual mult mai mare decât dimensiunea șablonului. Prin stabilirea pixelului adaptiv în poziția precedentă pe linia scanată curent în grila de tonuri diluate, se poate obține un context semnificativ mai eficace pentru codare.

Odată cu pixelii de pe nivelul care este codat, șabloanele de pe nivelurile diferențiale conțin pixeli în poziții corespunzătoare din nivelurile de rezoluție mai joasă. Transmisia progresivă lucrează în cealaltă direcție față de reducerea rezoluției, deoarece nivelurile de rezoluție mai scăzută sunt cunoscute atunci când cele de rezoluție mai înaltă sunt transmise, în timp ce, cu reducerea rezoluției, nivelurile mai înalte sunt cunoscute și cele mai joase sunt calculate pe baza lor. Patru șabloane sunt utilizate pentru transmisia progresivă, depinzând de faza pixelului codat, așa cum este arătat în figura 12. În fiecare caz, șablonul include patru pixeli ai nivelului de mai joasă rezoluție, marcați prin cercuri negre și șase pixeli ai nivelului curent, cinci dintre ei fiind marcați cu pătrate negre, iar al șaselea printr-un pătrat gri. În figura 12, ca și în figura 11, pixelii ale căror valori nu sunt cunoscute încă sunt hașurați deschis și fără contur întărit. Sunt 10 pixeli în aceste șabloane și patru faze; prin urmare $2^{12} = 4096$ contexte diferite sunt întreținute de către codor și decodor.

Pixelul gri în fiecare parte din figura 12 este un pixel „adaptiv". Ca și pixelul adaptiv din șabloanele nivelului de bază, poziției sale îi este permisă schimbarea, pe măsură ce imaginea este procesată. Figura 12 arată poziția sa implicită.

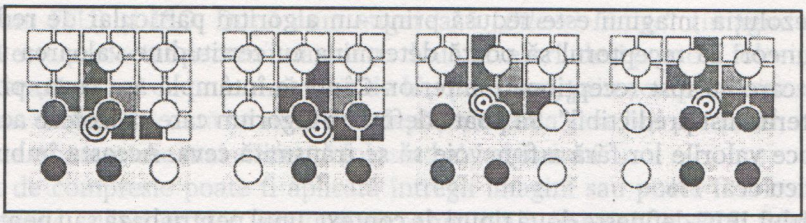


Figura 12. Şabloanele de nivel diferenţial în standardul JBIG

În conformitate cu descrierea de mai sus a modelelor de şablon, pentru fiecare din cele 4096 contexte diferite, două numărări asociate pot fi stocate – una pentru pixelii ţintă negri şi una pentru cei albi. Oricum, JBIG nu reprezintă numărările în mod explicit, ci le cufundă în interiorul însuşi al procesului de codare aritmetică, în special din raţiuni de implementare eficientă, aşa cum s-a arătat în subcapitolul 2.8. Statistica pixelilor diferă adeseori de la o regiune a imaginii la alta şi proiectarea mecanismului JBIG de estimare a probabilităţii asigură urmărirea probabilităţilor de schimbare, cu o rată mulţumitoare. Această rată este controlată de nivelul de precizie în probabilitatea celui mai puţin probabil simbol (LPS – *Less Probable Symbol*), care guvernează la rândul său mărimea tabelelor. Cu cât precizia este mai scăzută, cu atât mai repede răspunde estimatorul la o schimbare de probabilitate. În schimb, cu cât precizia este mai mare, cu atât mai bine lucrează codul, dacă statisticile sunt staţionare. Este deci un compromis între răspunsul în situaţii ne-staţionare şi eficienţa codării în situaţiile staţionare.

Avantajul acestei codări şi a tehnicii de estimare a probabilităţii este faptul că este foarte rapidă. Nici o multiplicare nu este implicată în codarea aritmetică. Estimarea este doar invocată atunci când re-normalizarea este necesară – cu alte cuvinte când este generat un bit supus compresiei – şi chiar şi atunci, este condus pe bază de tabel şi cere doar un index şi acces la memorie. Şi, în pofida aproximaţiilor implicate, se pierde foarte puţin din eficienţa codării.

4.5. *Formatul GIF pentru compresia fără pierderi a imaginii*

Probabil cel mai uzual format de compresie fără pierderi a imaginii utilizat în prezent este GIF (GIF – *Graphics Interchange Format*). Acesta a fost introdus de CompuServe în 1987, cu scopul de a minimiza timpul de încărcare a fişierelor prin legăturile prin modem; a fost dedicat ca mediu de schimb pentru informaţii grafice care ar fi putut fi afişate pe o varietate de platforme grafice hardware.[d]

GIF se aplică imaginilor în care fiecare pixel este reprezentat de opt biţi sau mai puţin. Codul pentru un pixel este un index într-un tabel care specifică o hartă a culorilor pentru întreaga imagine. Prin urmare, cu descriptorii de opt biţi per pixel se pot oferi doar 256 de culori posibile pentru fiecare pixel de imagine. Oricum, harta de culori

[d] O problemă cu formatul GIF care a apărut numai recent este că schema de compresie LZW pe care o foloseşte a fost patentată de Unisys şi producătorii care oferă produse bazate pe GIF trebuie să plătească licenţa. Un nou format de imagine PNG (*Portable Network Grafics*) este dezvoltat special pentru domeniul public; pentru detalii vezi `http://quest.jpl.nasa.gov/PNG`. PNG mai are în plus faţă de GIF avantajul că poate deservi imagini cu mai mult de opt biţi pe pixel.

poate să conțină culori alese dintr-o paletă mult mai largă. Formatul GIF permite tabelului de culori să fie specificat pentru fiecare imagine, sau pentru un grup de imagini care împart aceeași hară de culori, sau să se omită complet harta de culori. Dacă este inclusă, ea formează un prefix necomprimat al imaginii și poate specifica 256 de culori de intrare pentru tabel, fiecare de 24 de biți, câte opt pentru fiecare culoare primară, roșu, verde și albastru. Tabelul de culori nu este comprimat.

Schema de compresie folosită pentru secvența de valori de pixeli nu este croită deloc pentru imagini: este schema LZW care este destinată compresiei textului. Să ne amintim din subcapitolul 3.2 că LZW este o variantă a LZ78, metoda de compresie Ziv-Lempel bazată pe dicționar, care inițializează dicționarul pentru a conține un alfabet și apoi analizează fraze succesive din șirul de intrare, fiecare frază fiind găsită în dicționar, la fiecare etapă adăugând în dicționar fraza mărită cu un caracter adițional.

Să presupunem că lucrăm cu descriptori de opt biți pentru pixeli. Atunci schema de codare GIF inițializează dicționarul cu 256 de coduri de pixeli posibile – deși aceasta este numai noțional; nu este necesar să se încarce dicționarul cu aceste valori – și cu două coduri suplimentare, un cod „*clear*" (de blancare) și un cod „sfârșit-de-informație" și pornește apoi să codeze LZW secvența de valori de pixeli în ordinea scanată a rastrului.

O caracteristică adițională este inclusă pentru a face ușoară sărirea rapidă a unei imagini într-un fișier de imagini – deoarece un fișier poate conține câteva imagini. Informația codată LZW este grupată în blocuri de până la 256 de octeți (bytes), fiecare precedat de un numărător de octeți. Aceasta înseamnă că imaginea poate fi sărită fără a fi necesară decompresia.

4.6. *Schema FELICS pentru compresia imaginilor cu scară de gri*

FELICS, un acronim pentru *Fast, Efficient, Lossless Image Compression System*, este o tehnică simplă și remarcabil de eficientă de compresie fără pierderi a imaginilor cu scară de gri, care este comparabilă cu varianta fără pierderi a JPEG.[82,83] Ideea este de a coda fiecare pixel pe baza celor doi vecini ai săi și folosind o schemă non-adaptivă, special croită, pentru a reprezenta valoarea lui. Testată pe un set extins de imagini de test Landsat și pe câteva imagini generale, schema a fost catalogată ca oferind o compresie ușor mai ridicată decât modurile fără pierderi JPEG (versiunea codată aritmetic) și cu viteză considerabil mai mare, cel puțin pe arhitecturi de uz general. Oricum, nu funcționează bine pentru imagini foarte comprimabile și în particular nu poate sub nici o formă scădea sub 1 bit per pixel, indiferent cât de redundantă este imaginea.

Când o imagine este transmisă în ordinea de scanare a rastrului, fiecare pixel are doi vecini apropiați ale căror valori sunt cunoscute, situație ilustrată în figura 13(a). Pentru pixelii din interior, aceștia sunt cel de deasupra și cel de la stânga. Pentru pixelii de pe rândul de sus, ei sunt cei doi precedenți. Pentru pixelii din marginea stângă a imaginii, ei sunt primii doi de pe linia precedentă (o alternativă ar mai fi utilizarea primilor pixeli din cele două linii precedente). Primii doi pixeli din imagine, care nu se încadrează în aceste reguli, pot fi codați folosind o reprezentare binară standard cu un impact neglijabil asupra performanțelor.

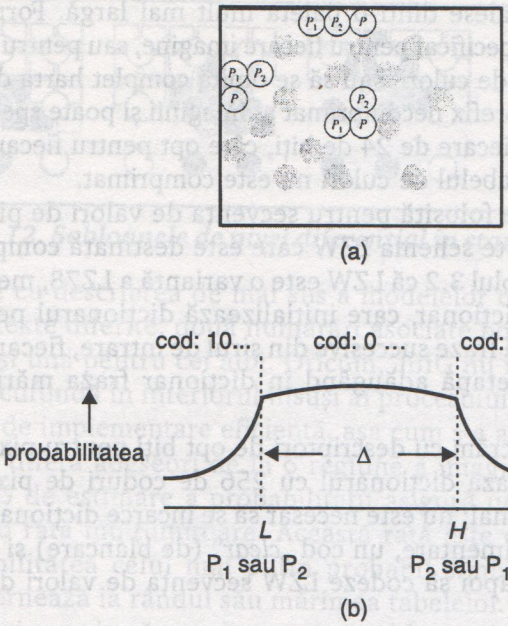


Figura 13. Cum depinde regiunea de codare a unui pixel de vecinii lui: (a) definirea vecinilor unui pixel; (b) cele trei regiuni de codare

Să presupunem că pixelul P, cu vecinii P_1 și P_2, este codat. Supraîncărcând această notație, o vom folosi pentru a reprezenta valorile scării de gri ale acestor pixeli. Sunt trei cazuri: valoarea lui P se află între celelalte două, sau se află deasupra celei mai mari, H, sau se află sub cea minimă L. Aceste trei situații sunt ilustrate schematic în figura 13(b), unde $L = \min\{P_1, P_2\}$ și $H = \max\{P_1, P_2\}$. În mod tipic, P se află în regiunea centrală aproximativ în jumătate din cazuri și aceasta poate fi convențional notată într-un bit. Un caz special poate apărea atunci când pixelii vecini P_1 și P_2 au valori de gri egale. Atunci, probabilitatea ca P să aibă aceeași valoare tinde să fie cumva mai mică de jumătate, deoarece este disponibilă numai o singură valoare și este avantajos să lărgim artificial regiunea centrală pentru a include, să zicem, trei niveluri de gri în loc de unul. Când P se află în afara regiunii centrale, situațiile peste / sub sunt simetrice și ele pot fi distinse printr-un al doilea bit.

Figura 13(b) arată distribuția lui P relativ la L și H. Când P este în regiunea centrală, valorile lui sunt distribuite aproape uniform, astfel că o codare binară standard dă în acest caz o compresie aproape optimală. Când este în afara regiunii centrale, probabilitatea scade destul de abrupt pe măsură ce valoarea se îndepărtează, în acest caz o alegere bună fiind codarea Golomb descrisă în subcapitolul 2.3. Vom considera fiecare dintre aceste situații puțin mai departe.

Definim $\Delta = H - L$ ca fiind dimensiunea regiunii centrale, așa cum este marcat în figura 13(b). Pentru a coda un pixel P între P_1 și P_2, distanța (*offset*) $P - L$ este codată în intervalul $[0, \Delta]$. Se folosește codarea binară obișnuită, cu simpla modificare descrisă la începutul subcapitolului 2.3 în cazul în care $\Delta + 1$ nu este o putere a lui doi.

Acum să considerăm utilizarea codurilor Golomb pentru reprezentarea valorilor de sub și peste regiunea centrală. Să ne amintim din subcapitolul 2.3 că, codurile Golomb au un parametru b și numărul $x > 0$ este codat ca $q + 1$ în unar, unde $q = \lfloor (x - 1)/b \rfloor$, urmat de $r = x - qb - 1$ codat în binar, cerând fie $\lfloor \log b \rfloor$ fie $\lceil \log b \rceil$ biți. Se pierde foarte puțin din eficiența de compresie prin folosirea cazului special de codare Golomb în care b este impus ca o putere a lui 2, $b = 2^m$. Atunci, partea binară este formată din cei m biți cei mai puțin semnificativi ai lui x, iar partea unară este formată din biții cei mai semnificativi.

Codarea Golomb funcționează cel mai bine atunci când m este ales astfel încât să se potrivească cel mai bine distribuției particulare care este codată. Pare rezonabil să se folosească Δ ca un context pentru a determina care valoare a lui m să fie folosită. Pentru a optimiza pe m pentru fiecare context, se poate păstra o înregistrare a fiecărei valori realizate în trecut. Pentru imagini cu o scară de gri de 8 biți sunt numai 256 de valori posibile pentru Δ și experiența arată că este arareori necesar să se folosească orice altceva decât $m = 0, 1, 2$ și 3, ceea ce corespunde parametrilor $b = 1, 2, 4$ și 8.

Este o treabă simplă să se mențină un tabel care să înregistreze, pentru fiecare valoare a lui Δ și m, lungimea cumulativă totală a codului care ar fi rezultat dacă parametrul m s-ar fi folosit în contextul Δ. Parametrul cu lungimea codului cumulativ cel mai mic este ales pentru a coda următoarea valoare întâlnită în context. Desigur, decodorul menține același tabel, astfel încât poate decoda cu succes codurile Golomb. În acest fel se folosește o valoare de parametru corespunzătoare pentru fiecare Δ și foarte puțin surplus este necesar pentru a menține informația dorită.

4.7. *Schema CALIC pentru compresia imaginilor cu tonuri continue*

CALIC (= *Context-based Adaptive Lossless Image Codec*) este recent intrat în rândul compresoarelor de imagine fără pierderi.[84,85] Acesta a fost confirmat ca întrecând toate celelalte scheme fără pierderi și a fost catalogat în fruntea a nouă scheme de compresie fără pierderi evaluate de ISO / JPEG în iulie 1995. Deși este capabilă de o implementare eficientă, CALIC este conceptual o schemă foarte elaborată și vom schița aici numai ideile principale. O caracteristică practică importantă este aceea că implementarea are nevoie de numai atâta spațiu de stocare cât să păstreze trei rânduri de imagine și încă puțină memorie care nu depinde de dimensiunea imaginii.

CALIC codează o imagine în ordinea firească de scanare a rastrului și își bazează toate calculele pe o mică vecinătate de doisprezece pixeli care îl preced pe cel curent în ordinea de transmisie. Ca și FELICS, el prezice o valoare a pixelului bazată pe cele ale pixelilor din vecinătate și apoi transmite o corecție care reprezintă discrepanța între valoarea adevărată a pixelului și cea prezisă. Oricum, atât mecanismul de predicție cât și medierea erorilor de transmisie sunt mult mai elaborate decât la FELICS.

Pentru a prezice o valoare de pixel, se folosește o estimare locală a gradientului pixelului, din pixelii vecini, din patru direcții, orizontal, vertical și în direcțiile ambelor diagonale. Aceste estimări sunt folosite pentru a detecta magnitudinea și orientarea marginilor (contururilor) în imaginea de intrare, astfel ca să se poată face ajustări ale predicției în consecință. Se disting opt cazuri diferite, depinzând de detectarea

marginilor abrupte sau non-abrupte şi pentru fiecare caz se foloseşte altă formulă de predicţie. Ajustarea predicţiei în funcţie de gradient este o încercare explicită de a determina linii şi contururi în imagini.

Odată ce s-a obţinut predicţia „reglată de gradient", se aplică o corecţie bazată pe şablonul particular pe care îl formează pixelii învecinaţi. Aceasta este necesară deoarece gradientul singur nu poate caracteriza adecvat unele dintre cele mai complexe legături între pixelul prezis şi vecinătatea lui. Modelarea contextuală poate exploata structuri de ordin mai înalt, cum ar fi modelele de texturi din imagine. Aceasta se face prin cuantizarea vecinătăţii în câteva sute de clase de condiţionări şi estimarea erorii de predicţie în fiecare din aceste clase separat. Pentru a ne asigura că numărătorile de eşantioane sunt suficient de mari pentru a obţine o bună estimare şi pentru a reduce timpul şi complexitatea spaţială, sunt estimate aşteptările condiţionale în locul funcţiilor de densitate condiţională. Este apoi folosită eroarea aşteptării condiţionale estimate pentru a corecta predicţia „reglată de gradient". Efectul net este un predictor neliniar, bazat pe context, care se autocorectează prin adaptarea la erorile făcute în acelaşi context în trecut.

În sfârşit, diferenţa între valoarea adevărată a pixelului şi valoarea corectată a predicţiei este codată entropic. În comanda codorului entropic, distribuţia de probabilitate a erorii de predicţie este estimată cu (numai) opt clase de condiţii diferite. Aceste clase de condiţii permit schemei să se adapteze la diferite modele de texturi spaţiale. Clasa condiţională cea mai potrivită este determinată printr-o estimare a predictibilităţii semnalului în acel punct. Această predictibilitate este măsurată printr-o funcţie liniară de gradientele vertical şi orizontal şi prin eroarea de predicţie de la pixelul precedent. Coeficienţii optimali pot fi calculaţi folosind regresia liniară *off-line* pentru a ajunge la o îmbunătăţire a eficienţei de codare.

Dacă această etapă de corecţie finală este omisă, rezultă o schemă de compresie cu pierderi, care este comparabilă cu standardul de compresie cu pierderi JPEG: de fapt, se atinge o compresie semificativ mai mare decât cu JPEG, pentru aceeaşi măsură obiectivă a calităţii.

O altă inovaţie importantă a CALIC este că distinge regiunile binare (cu treceri bruşte) ale imaginii de cele cu tonuri continue, pe o bază locală, în loc de o bază globală. El face acest lucru prin examinarea a şase pixeli precedenţi locali: dacă ei nu au mai mult de două valori diferite (ceea ce nu înseamnă neapărat alb şi negru), pixelul care urmează este codat în mod binar, altfel este codat folosind metoda predictivă descrisă mai sus. În modul binar, vecinătatea celor şase pixeli este folosită pentru a condiţiona distribuţia valorii pixelului care urmează, aşa cum a fost descris în subcapitolul 4.2. Există, desigur, rezerva de „scăpare" la o valoare de pixel, care este diferită de a celor două care apar în vecinătate (introducerea unui „escape").

4.8. *Performanţele metodelor de compresie fără pierderi*

Pentru a evalua performanţele relative ale metodelor decrise mai sus de compresie fără pierderi a imaginii, s-au realizat teste asupra unor implementări standard. Algoritmii de compresie a imaginii sunt foarte sensibili la un număr de factori şi aceste teste nu sunt proiectate ca o comparaţie comprehensivă ci doar ca o indicaţie aproximativă a performanţelor diferitelor metode.

Imaginile folosite pentru compresia cu două niveluri au fost opt imagini de test CCITT care au fost introduse acum mulți ani, pentru a evalua performanțele standardelor Group 3 și 4 pentru facsimil;[73] acestea conțin un total de 33 milioane de pixeli de 1 bit. Ele sunt considerate reprezentative pentru documente de birou. Oricum, performanțele algoritmilor de compresie sunt foarte sensibile la tipul de imagine care este comprimată. De exemplu, codurile statice ale metodelor facsimil Group 3 și 4 vor *expanda* detaliile bi-nivel ale imaginii și astfel aceste două scheme includ un semnalizator (indicator, flag) „escape" care indică faptul că o linie scanată dată se transmite necomprimat.

Schemele de compresie testate au fost implementările disponibile public ale Group 3, Group 4, JBIG și GIF bi-nivel. Imaginile JBIG au fost comprimate folosind codarea secvențială fără niveluri diferențiale; aceasta este destinată a asigura cea mai bună compresie. Oricum, versiunea de codare Huffman a JBIG a fost folosită deoarece modulul de codare aritmetică nu este disponibil în implementări de domeniu public.

Figura 14 arată rezultatele. Raportul de compresie este reprezentat pe orizontală; de exemplu raportul de compresie cel mai bun este în jurul a 20:1 pentru metoda JBIG, iar GIF asigură o compresie puțin mai slabă decât CCITT Group 3. Viteza este reprezentată vertical, în milioane de pixeli pe secundă pe un Sparc Center 1000 Sun și vitezele de codare și decodare sunt foarte apropiate (în limitele a 10% una față de alta); graficul arată mediile. Compresia GIF bi-nivel este extrem de lentă: acesta este un neajuns al programului pe care l-am folosit, care presupune că imaginea este color și, în naivitatea lui, caută o paletă bună în tabelul de culori. Pentru celelalte trei metode, compromisul între compresie și viteză este evident. Este de așteptat ca, în timp ce implementarea JBIG este destul de optimizată pentru viteză, celelalte să poată fi îmbunătățite considerabil dacă s-ar acorda o mai mare atenție detaliilor de codare.

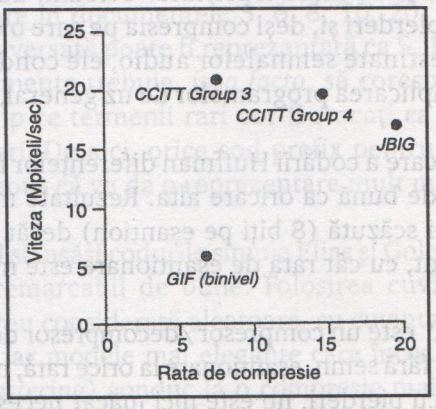


Figura 14. Performanțele metodelor de compresie bi-nivel

Pentru a compara schemele de compresie pentru tonuri continue, s-au folosit nouă imagini, totalizând 2,2 milioane de pixeli de opt biți. Algoritmii testați au fost implementările standard GIF, FELICS și CALIC. Figura 15 arată rezultatele. Conform așteptărilor, CALIC dă o compresie puțin mai bună decât FELICS, deși cu o scădere apreciabilă a vitezei. GIF este mai slabă decât celelalte două metode. S-a avut grijă de codarea fiecăreia dintre aceste implementări și astfel comparațiile mai precise cu privire la viteză reflectă mai mult complexitatea intrinsecă a metodei decât în cazul bi-nivel.

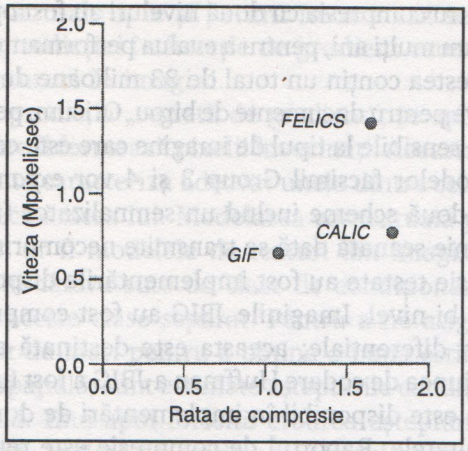


Figura 15. Performanțele metodelor de compresie a tonurilor de gri

5. Alte forme de date

În acest capitol vom descrie pe scurt câteva alte aplicații ale metodelor de compresie fără pierderi.

5.1. *Compresia sunetului*

Cele mai multe scheme de compresie a formelor de undă audio sunt cu pierderi și aceasta este în mod real cea mai potrivită metodă de compresie audio – chiar dacă se urmăresc rezultate de mare fidelitate. Oricum, au fost dezvoltate scheme de compresie audio fără pierderi și, deși compresia pe care o obțin este mult inferioară schemelor cu pierderi destinate semnalelor audio, ele conduc la rezultate mai bune decât cele obținute prin aplicarea programelor de uz general, de compresie a fișierelor audio.

În general, simpla aplicare a codării Huffman diferențelor între eșantioane succesive pare a fi o metodă la fel de bună ca oricare alta. Rezultate mult mai bune se obțin cu semnale de o calitate mai scăzută (8 biți pe eșantion) decât cu calitate mai bună (16 biți pe eșantion); mai mult, cu cât rata de eșantionare este mai mare, cu atât crește și rata de compresie.

De exemplu, VOCPACK[e] este un compresor /decompresor de sunet digital de 8 biți. El poate comprima date cu și fără semn, eșantionate la orice rată, mono sau stereo. Deoarece metoda utilizată nu este cu pierderi, nu este nici măcar necesar să se curețe headerele fișierului înainte de compresie. Deși a fost dezvoltat pentru a fi folosit cu fișiere SoundBlaster (.VOC) și Windows (.WAV), orice fișier de sunet cu 8 biți pe eșantion poate fi comprimat, deoarece nu există nici o condiție asupra structurii fișierului. Ratele de compresie obținute (fracțiunea din spațiul original necesar pentru reprezentarea comprimată) variază tipic între 80% din mărimea originală pentru fișiere eșantionate la 11 kHz și 40% pentru 44 kHz.

[e] Disponibil la `ftp://oak.oakland.edu/simtel/msdos/sound/vocpak20.zip`.

5.2. *Compresia indexurilor*

Devine din ce în ce mai mult o practică uzuală ca mari cantități de text să fie disponibile on-line (direct) pentru căutare și browsing (răsfoit). Într-adevăr, cei mai mulți utilizatori de calculatoare își creează repede propriul spațiu de informații personale ocupând zeci sau sute de megabytes (megaocteți) de stocare în care depun documente și articole de poștă electronică. Pe scena comercială este uzual ca aceste baze de informații să se extindă pe arii de mai mulți gigabytes (gigaocteți).

Compresia are o aplicație clară în stocarea acestor depozite și tehnicile descrise în capitolul 3 pot fi adaptate pentru aceasta. Desigur, folosirea compresiei poate economisi în unele cazuri atât spațiu cât și timp – este posibil ca pentru o unitate CPU, costul de decompresie a unui document comprimat să fie mai mare decât câștigul obținut prin reducerea costului de încărcare a documentului în stare comprimată.[13]

Compresia poate fi de asemenea utilă pentru a facilita stocarea indexurilor care sunt necesare dacă trebuie realizată o căutare eficientă. Un index inversat pentru o colecție de documente stochează, pentru fiecare cuvânt (și număr) care apare oriunde în colecție o listă de identificatori care indică documentul în care apare acel cuvânt. Stocat într-un mod oarecare, un astfel de index poate ocupa o fracțiune substanțială din spațiul ocupat de colecția însăși, de vreme ce fiecare cuvânt dintr-un text de limbă engleză corespunde în medie la 4-6 caractere, iar un identificator întreg de 32 de biți pentru un document ocupă de asemenea 4 bytes (octeți).

Există, oricum, o mare structurare într-un astfel de index inversat. Dacă se presupune că fiecare listă de identificatori ai unui document pot fi stocați într-o ordine crescătoare, atunci fiecare listă poate fi stocată ca o secvență de goluri (pauze) în document, mai degrabă decât o listă de numere absolute de documente. De exemplu, dacă cuvântul *pisică* apare în documentele 5, 6, 8, 10, 20, 21 și 31 dintr-o anumită colecție, atunci lista lui inversată poate fi reprezantată ca 5, 1, 2, 2, 10, 1, 10. Termenii care apar în multe documente trebuie, *ipso facto*, să corespundă listelor care conțin multe pauze mici, în timp ce termenii rari vor fi stocați ca un număr relativ mic de posibile distanțe mai mari. De aici, orice cod prefix pentru întregi, care favorizează valori mici, este de presupus că va da o reprezentare mult mai compactă decât un cod binar plat.

Acesta este chiar cazul și însăși codurile simple Elias și Golomb, descrise în subcapitolul 2.3 dau compresii remarcabil de bune. Folosirea cuvintelor într-o colecție de documente poate fi cu greu considerată aleatoare, cu cuvinte care intră și ies din uz pe măsură ce trece timpul, iar modele mai elegante care încearcă să exploateze efectele de compartimentare (*clustering*) conduc la o compresie mai bună.[86] Pentru o colecție tipică de documente, adunând articole cu lungimi de pagini de câțiva kiloocteți, este posibil să se reprezinte informația din index în aproximativ 5-10% din spațiul ocupat de textul care este indexat, ceea ce este o economie remarcabilă.[86-94]

5.3. *Imagini textuale*

Cei mai mulți algoritmi de compresie de imagine sunt de uz general, aplicându-se deopotrivă oricărui fel de imagine. Oricum, sunt multe situații, cum este cazul facsimilului și a aplicațiilor de arhivare a documentelor, în care sunt codate imaginile bi-nivel care

conţin în principal text tipărit; le numim *imagini textuale*. Se poate obţine o compresie excelentă pe imaginile textuale ţinând cont de natura lor. O opţiune este de a realiza o recunoaştere optică a caracterelor pe text şi de a transmite codurile ASCII ale caracterelor, împreună cu unele informaţii privind poziţiile lor pe pagină. Problema cu această metodă este că recunoaşterea caracterelor nu este de încredere totală şi unele greşeli pot fi catastrofice. O alternativă este de a identifica tiparele repetate din imagine, de obicei caractere, şi de a le înlocui cu pointere într-o bibliotecă de tipare.[95] În general procesul conţine următorii cinci paşi.

1. Găsirea, izolarea şi extragerea tuturor *mărcilor* din imagine, care sunt grupuri de pixeli negri conectaţi.
2. Construirea unei biblioteci care să conţină toate mărcile diferite care apar.
3. Identificarea simbolului din bibliotecă corespunzător fiecarei mărci din imagine şi măsurarea distanţei (offset) dintre coordonatele unei mărci şi cele ale următoarei.
4. Compresia şi transmiterea bibliotecii, a secvenţei simbolurilor şi a offseturilor.

Din informaţia de la pasul patru, se poate extrage o aproximaţie a imaginii originale numită *reconstrucţia textului*. Aceasta dă o metodă de compresie cu pierderi. Pentru a reface textul fără pierderi, ceea ce poate fi necesar pentru scopuri legale, medicale, sau istorice, un al cincilea pas se mai poate adăuga.

5. Transmiterea unei informaţii adiţionale suficiente pentru a reface imaginea originală din textul reconstruit.

Combinaţia celor două moduri (cu şi fără pierderi) conduce la o procedură în două etape pentru transmisia progresivă, care este în mod particular atractivă pentru aplicaţiile practice care implică răsfoirea (*browsing*) în baze de date mari. În modul cu pierderi, schema depăşeşte alte scheme de compresie cu pierderi a imaginii şi totuşi păstrează o redare a imaginii foarte clară, de bună calitate. În schema fără pierderi, depăşeşte alte scheme fără pierderi cum ar fi JBIG.

5.4. *Compresia fişierelor în sistem*

Sistemele de compresie pe disc „*on-the-fly*" interceptează datele care urmează a fi scrise pe disc şi le comprimă înainte de a fi stocate. Din punctul de vedere al utilizatorului, durează ceva mai mult scrierea şi citirea datelor, dar discul stochează considerabil mai multe date decât înainte. Este esenţial ca aceste sisteme să fie fără pierderi. Ele trebuie să fie de asemenea foarte rapide pentru a nu avea un impact serios asupra performanţelor sistemului – de fapt, în unele cazuri ele aduc îmbunătăţiri deoarece mai puţine date sunt transferate de pe şi pe disc.

Metodele bazate pe LZ77 sunt ideale pentru aceste aplicaţii deoarece ele codează şi decodează foarte rapid, prin folosirea unei structuri de căutare foarte eficiente a datelor, cum este tabelul combinat (*hash table*) şi prin stabilirea unor limite de căutare. Limitarea căutării înseamnă că unele potriviri bune pot fi sărite în detrimentul compresiei, dar acesta este un compromis rezonabil pentru a face sistemul cât se poate de transparent. Se poate înbunătăţi compresia prin recomprimarea fişierelor atunci când procesorul este în repaus (*idle*), folosind o căutare mai aprofundată a potrivirilor. Algoritmul de decompresie este identic, indiferent cum au fost găsite potrivirile, astfel că recompresia este un proces ideal de „background".

Un sistem de compresie a fişierelor este în particular vulnerabil la erori şi de aceea nivelul de încredere este o preocupare majoră. Datele sunt stocate în mod normal pe disc, în blocuri de dimensiuni fixe, dar compresia face dimensiunile unui bloc imprevizibile, ceea ce complică controlul fişierului. Mai mult, o eroare mică într-un fişier comprimat îl poate face de necitit. Prin urmare, probleme minore care ar putea să nu afecteze serios un fişier necomprimat, pot avea un impact major asupra unuia comprimat. În consecinţă, cele mai bune compresoare – sistem pentru fişiere sunt acelea care sunt rezistente la erori, mai mult decât cele care dau cea mai bună compresie – evitarea erorilor pe disc este mult mai importantă pentru utilizator decât înghesuirea unui mic procent suplimentar de date pe el.

6. Concluzie

Compresia fără pierderi este potrivită pentru multe tipuri de date. În particular, textele şi unele forme de imagine trebuie stocate exact şi aproximaţii nu pot fi tolerate. În acest articol am aruncat o privire de ansamblu asupra unui spectru larg de tehnici de compresie fără pierderi, incluzând o varietate de metode de codare şi modele de compresie potrivite pentru text şi imagini bi-nivel, cu scară de gri şi tip document, precum şi baze de date tip index.

Software

Multe dintre tehnicile descrise în acest articol au fost implementate de autorii articolului în colaborare cu J. Zobel (RMIT, Australia). Un sistem de compresie şi refacere a imaginii şi a documentelor – numit MG – a fost făcut disponibil public şi poate fi copiat de la `ftp://munnari.oz.au/pub/mg`. Witten *et* al.[93] descriu diverse componente ale acestui sistem şi dau o introducere tip manual a facilităţilor pe care le oferă.

Mulţumiri

Această lucrare a fost susţinută de Australian Research Council şi de Natural Science and Engineering Research Council of Canada. Suntem recunoscători lui Stuart Inglis care a efectuat testele descrise în subcapitolul 4.8.

Referinţe bibliografice

1. J. Rissanen şi G. G. Langdon, "Universal modeling and coding", *IEEE Trans. on Information Theory* **IT-27**, 1 (1981) 12–23.
2. G. G. Langdon, "A note on the Ziv–Lempel model for compressing individual sequences", *IEEE Trans. on Information Theory* **IT-29**, 2 (1983) 284–287.
3. T. C. Bell şi I. H. Witten, "The relationship between greedy parsing and symbolwise text compression", *J. ACM* **41**, 4 (1994) 708–724.
4. C. E. Shannon, "A mathematical theory of communication", *Bell Systems Technical J.* **27** (1948) 379–423, 623–656.
5. P. Elias, "Universal codeword sets and representations of the integers", *IEEE Trans. on Information Theory* **IT-21**, 2 (1975) 194–203.

6. S. W. Golomb, "Run-length encodings", *IEEE Trans. on Information Theory* **IT-12**, 3 (1966) 399–401.
7. R. G. Gallager şi D. C. Van Voorhis, "Optimal source codes for geometrically distributed integer alphabets", *IEEE Trans. on Information Theory* **IT-21**, 2 (1975) 228–230.
8. R. F. Rice, "Some practical universal noiseless coding techniques", Technical Report 79–22. Jet Propulsion Laboratory, Pasadena, California, 1979.
9. D. A. Huffman, "A method for the construction of minimum-redundancy codes", *Proc. Inst. Radio Engineers* 40, 9 (1952) 1098–1101.
10. E. S. Schwartz şi B. Kallick, "Generating a canonical prefix encoding", *Commun. ACM* **T**, 3 (1964) 166–169.
11. J. B. Connell, "A Huffman-Shannon-Fano code", *Proc. IEEE* **61**, 7 (1973) 1046–1047.
12. D. Hirschberg şi D. Lelewer, "Efficient decoding of prefix codes", *Commun. ACM* **33**, 4 (1990) 449–459.
13. J. Zobel şi A. Moffat, "Adding compression to a full-text retrieval system", *Software–Practice and Experience* **25**, 8 (1995) 891–903.
14. A. Moffat şi A. Turpin, "On the implementation of minimum-redundancy prefix codes", versiune preliminară înaintată către *Proc. IEEE Data Compression Conference,* martie 1996, pp. 170–179.
15. Y. Choueka, S. T. Klein şi Y. Perl, "Efficient variants of Huffman codes in high level languages", în *Proc. 8th ACM-SIGIR Conf. on Information Retrieval,* Montreal, Canada, 1985, pp. 122–130.
16. A. Sieminski, "Fast decoding of the Huffman codes", *Information Processing Lett.* **26**, 5 (1988) 237–241.
17. J. van Leeuwen, "On the construction of Huffman trees", în *Proc. 3rd Int. Colloquium on Automata, Languages, and Programming,* Edinburgh University, Scotland, 1976, pp. 382–410.
18. A. Moffat şi J. Katajainen, "In-place calculation of minimum-redundancy codes", în *Proc. Workshop on Algorithms and Data Structures,* redactori S. G. Akl, F. Dehne şi J.-R. Sack, LNCS 955, Kingston University, Canada, 1995, pp. 393–402.
19. L. L. Larmore şi D. S. Hirschberg, "A fast algorithm for optimal length-limited Huffman codes", *J. ACM* **37**, 3 (1990) 464–473.
20. J. Katajainen, A. Moffat şi A. Turpin, "A fast and space-economical algorithm for length-limited coding", în *Proc. Int. Symp. on Algorithms and Computation,* redactori J. Staples, P. Eades, N. Katoh şi A. Moffat, LNCS 1004, Springer-Verlag, Cairns, Australia, 1995, pp. 12–21.
21. B. Schieber, "Computing a minimum-weight k-link path in graphs with the concave Monge property", în *Proc. 6th Ann. Symp. Discrete Algorithms,* San Francisco, California, 1995, pp. 405–411.
22. A. Moffat şi A. Turpin, "Efficient construction of minimum-redundancy codes for large alphabets", înaintat versiune preliminară către *Proc. IEEE Data Compression Conference,* martie 1995, pp. 192–201.
23. T. C. Hu şi A. C. Tucker, "Optimal computer search trees and variable length alphabetic codes", *SIAM J. of Applied Mathematics* **21** (1971) 514–532.
24. L. L. Larmore şi T. M. Przytycka, "Constructing Huffman trees in parallel", *SIAM J. on Computing* **24** (1995) 1163–1169.
25. A. Zandi, B. Iyer şi G. Langdon, "Sort order preserving data compression for extended alphabets", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1993, pp. 330–339.

26. J. Rissanen, "Generalised Kraft inequality and arithmetic coding", *IBM J. Research and Development* **20** (1976) 198–203.
27. J. Rissanen şi G. G. Langdon, "Arithmetic coding", *IBM J. Research and Development* **23**, 2 (1979) 149–162.
28. G. G. Langdon, "An introduction to arithmetic coding", *IBM J. Research and Development* **28**, 2 (1984) 135–149.
29. I. H. Witten, R. M. Neal şi J. G. Cleary, "Arithmetic coding for data compression", *Commun. ACM* **30**, 6 (1987) 520–541.
30. D. W. Jones, "Application of splay trees to data compression", *Commun. ACM* **31**, 8 (1988) 996–1007.
31. A. Moffat, "Linear time adaptive arithmetic coding", *IEEE Trans. on Information Theory* **36**, 2 (1990) 401–406.
32. P. M. Fenwick, "A new data structure for cumulative probability tables", *Software–Practice and Experience* **24**, 3 (1994) 327–336; Errata publicată în **24**, 7 (1994) 677.
33. A. Moffat, N. Sharman, I. H. Witten şi T. C. Bell, "An empirical evaluation of coding methods for multi-symbol alphabets", *Information Processing & Management* **30**, 6 (1994) 791–804.
34. J. Rissanen şi K. M. Mohiuddin, "A multiplication-free multialphabet arithmetic code", *IEEE Trans. on Communications* **37**, 2 (1989) 93–98.
35. D. Chevion, E. D. Karnin şi E. Walach, "High efficiency, multiplication free approximation of arithmetic coding", în *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, 1991, pp. 43–52.
36. G. Feygin, P. G. Gulak şi P. Chow, "Minimizing excess code length and VLSI complexity in the multiplication free approximation of arithmetic coding", *Information Processing & Management* **30**, 6 (1994) 805–816.
37. Moffat, R. M. Neal şi I. H. Witten, "Arithmetic coding revisited", în *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, 1995, pp. 202–211. Software disponibil la adresa ftp://munnari.oz.au/pub/arith-coder.
38. R. G. Gallager, "Variations on a theme by Huffman", *IEEE Trans. on Information Theory* **IT-24**, 6 (1978) 668–674.
39. G. V. Cormack şi R. N. Horspool, "Algorithms for adaptive Huffman codes", *Information Processing Lett.* **18** (1984) 159–165.
40. D. E. Knuth, "Dynamic Huffman coding", *J. Algorithms* **6** (1985) 163–180.
41. J. S. Vitter, "Algorithm 673: Dynamic Huffman coding", *A CM Trans.* on *Mathematical Software* **15**, 2 (1989) 158–167.
42. W. W. Lu şi M. P. Gough, "A fast-adaptive Huffman coding algorithm", *IEEE Trans. on Information Theory* 41, 4 (1993) 535–538.
43. A. Bookstein şi S. T. Klein, "Is Huffman coding dead?", *Computing* **50**, 4 (1993) 279–296.
44. R. M. Capocelli, R. Giancarlo şi I. J. Taneja, "Bounds on the redundancy of Huffman codes", *IEEE Trans. on Information Theory* **IT-32** (1986) 854–857.
45. R. M. Capocelli şi A. De Santos, "New bounds on the redundancy of Huffman codes", *IEEE Trans. on Information Theory* **IT-37** (1991) 1095–1104.
46. D. Manstetten, "Tight upper bounds on the redundancy of Huffman codes", *IEEE Trans. on Information Theory* **38, 1** (1992) 144–151.
47. W. B. Pennebaker, J. L. Mitchell, G. G. Langdon şi R. B. Arps, "An overview of the basic principles of the Q-coder adaptive binary arithmetic coder", *IBM J. Research and Development* **82**, 6 (1988) 717–726.

48. W. B. Pennebaker şi J. L. Mitchell, "Probability estimation for the Q-coder", *IBM J. Research and Development* **32**, 6 (1988) 737–752.
49. J. G. Cleary şi I. H. Witten, "A comparison of enumerative and adaptive codes", *IEEE Trans. on Information Theory* **IT-30**, 2 (1984) 306–315.
50. J. Ziv şi A. Lempel, "A universal algorithm for sequential data compression", *IEEE Trans. on Information Theory* **IT-23**, 3 (1977) 337–343.
51. J. Ziv şi A. Lempel, "Compression of individual sequences via variable rate coding", *IEEE Trans. on Information Theory* **IT-24**, 5 (1978) 530–536.
52. J. A. Storer şi T. G. Szymanski, "Data compression via textual substitution", *J. ACM* **29** (1982) 928–951.
53. T. C. Bell, "Better OPM/L text compression", *IEEE Trans. on Communications* **COM-34** (1986) 1176–118-2.
54. E. R. Fiala şi D. H. Green, "Data compression with finite windows", *Communications ACM* **32**, 4 (1989) 490–505.
55. R. N. Horspool, "The effect of non-greedy parsing in Ziv–Lempel compression methods", în *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, 1995, pp. 302–311.
56. R. N. Williams, "An extremely fast Ziv–Lempel data compression algorithm", în *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, 1991, pp. 362–371.
57. T. A. Welch, "A technique for high performance data compression", *IEEE Computer* **17** (1984) 8–20.
58. P. C. Gutmann şi T. C. Bell, "A hybrid aproach to text compression", în *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, 1994, pp. 225–234.
59. D. T. Hoang, P. M. Long şi J. S. Vitter, "Multiple-dictionary compression using partial matching", în *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, 1995, pp. 272-281.
60. T. C. Bell, "A unifying theory and improvements for existing approaches to text compression", teză de doctorat, University of Canterbury, Christchurch, New Zealand, 1986.
61. J. G. Cleary şi I. H. Witten, "Data compression using adaptive coding and partial string matching", *IEEE Trans. on Commun.* **COM-82** (1984) 396–402.
62. I. H. Witten şi T. C. Bell, "The zero frequency problem: Estimating the probabilities of novel events in adaptive text compression", *IEEE Trans. on Information Theory* **37**, 4 (1991) 1085–1094.
63. A. Moffat, "Implementing the PPM data compression scheme", *IEEE Trans. on Communications* **38**, 11 (1990) 1917–1921.
64. P. G. Howard şi J. S. Vitter, "Practical implementations of arithmetic coding", în *Image and Text Compression*, Kluwer Academic, Norwell, Massachusetts, 1992.
65. J. G. Cleary, W. Teahan şi I. H. Witten, "Unbounded length contexts for PPM", în *Proc. IEEE Data Compression Conf.*, Snowbird, Utah, 1995, pp. 52–61.
66. M. Burrows şi D. J. Wheeler, "A block-sorting lossless data compression algorithm", Technical Report 124, Digital Equipment Corporation, Palo Alto, California, 1994.
67. C. E. Shannon, "Prediction and entropy of printed English", *Bell Systems Technical J.* **30** (1951) 55.
68. T. M. Cover şi R. C. King, "A convergent gambling estimate of the entropy of English", *IEEE Trans. on Information Theory* **IT-24** (1978) 413–421.
69. G. V. Cormack şi R. N. Horspool, "Data compression using dynamic Markov modeling", *The Computer J.* **30**, 6 (1987) 541–550.
70. J. Teuhola şi T. Raita, "Application of a finite-state model to text compression", *The Computer J.* **36**, 7 (1993) 607–614.
71. T. C. Bell şi A. Moffat, "A note on the DMC data compression scheme", *The Computer J.* **32**, 1 (1989) 16–20.

72. S. Bunton, "The structure of DMC", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1995, pp. 72–81.
73. R. Hunter şi A. H. Robinson, "International digital facsimile coding standards", *Proc. IEEE* **68,** 7 (1980) 854–867.
74. G. G. Langdon şi J. Rissanen, "A simple general binary source code", *IEEE Trans. on Information Theory* **IT-28**, 5 (1982) 800–803.
75. W. B. Pennebaker şi J. L. Mitchell, *JPEG: Still Image Data Compression Standard,* Van Nostrand Reinhold, New York, 1993.
76. W. K. Pratt, P. J. Capitant, W. H. Chen, E. R. Hamilton şi R. H. Wallis, "Combining symbol matching facsimile data compression system", *Proc. IEEE* **68**, 7 (1980) 786–796.
77. Y. Yasuda, Y. Yamazake, T. Kamae şi K. Kobayashi, "Advances in FAX", *Proc. IEEE* **73,** 4 (1985) 706–730.
78. D. Bodson, S. J. Urban, A. R. Deutermann şi C. E. Clarke, "Measurement of data compression in advanced group 4 facsimile systems", *Proc. IEEE* **73,** 4 (1985) 731–739.
79. P. Tischer, R. Worley, A. J. Maeder şi M. Goodwin, "Context-based lossless image compression", *The Computer J.* **36**, 1 (1993) 68–77.
80. A. Moffat, "Two-level context-based compression of binary images", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1991, pp. 382–391.
81. T. Yoshida, T. Endoh, N. Kawamura şi H. Kato, "Image reduction system", US Patent 5,159,468, 1992.
82. P. G. Howard şi J. S. Vitter, "Fast and efficient lossless image compression", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1993, pp. 351–360.
83. P. G. Howard, "The design and analysis of efficient lossless data compression systems", teză de doctorat, Brown University, Rhode Island, 1993; disponibilă ca Technical Report CS-93-28.
84. X. Wu, "Context selection and quantization for lossless image coding", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1995, p. 453.
85. X. Wu, N. Memon şi K. Sayood, "A context-based, adaptive, lossless/nearly-lossless coding scheme for continuous-tone images", document în lucru ISO/IEC SC29/WG1/N256, 1995.
86. A. Bookstein, S. T. Klein şi T. Raita, "Markov models for clusters in concordance compression", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1994, pp. 116–125,
87. M. Jakobsson, "Huffman coding in bit-vector compression", *Information Processing Lett.* **7,** 6 (1978) 304–307.
88. A. S. Fraenkel şi S. T. Klein, "Novel compression of sparse bit-strings–Preliminary report", în *Combinatorial Algorithms on Words,* redactori A. Apostolico şi Z. Galil, vol. 12, NATO ASI Series F, Springer-Verlag, Berlin, 1985, pp. 169–183.
89. Y. Choueka, A. S. Fraenkel şi S. T. Klein, "Compression of concordances in full-text retrieval systems", în *Proc. ACM-SIGIR Int. Conf. on Research and Development in Information Retrieval,* Grenoble, France, 1988, pp. 597–612.
90. A. Bookstein, S. T. Klein şi T. Raita, "Model based concordance compression", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1992, pp. 82–91.
91. A. Moffat şi J. Zobel, "Parameterised compression for sparse bitmaps", în *Proc. ACM-SIGIR Int. Conf. on Research and Development in Information Retrieval,* Copenhagen, 1992, pp. 274–285.
92. T. C. Bell, A. Moffat, C. G. Nevill-Manning, I. H. Witten şi J. Zobel, "Data compression in full-text retrieval systems", *J. of the American Society for Information Science* **44,** 9 (1993) 508-531.

93. I. H. Witten, A. Moffat şi T. C. Bell, *Managing Gigabytes: Cornpressing and Indexing Documents and Images,* Van Nostrand Reinhold, New York, 1994.
94. A. Moffat şi L. Stuiver, "Exploiting clustering in inverted file compression", în *Proc. IEEE Data Compression Conf.,* Snowbird, Utah, 1996, pp. 82–91.
95. I. H. Witten, T. C. Bell, H. Emberson, S. Inglis şi A. Moffat, "Textual image compression: Two-stage lossy/lossless encoding of textual images", *Proc. IEEE* **82,** 6 (1994) 878-888.
96. CCITT, Proiectul recomandării T.82 0 ISO DIS 11544: Coded Representation of Picture and Audio Information–Progressive Bi-Level Image Compression, 1993.

- **Compresia semnalelor:** *procesarea și codarea digitală a semnalelor; tendințe;*
- **Metode curente de codare a vorbirii:** *algoritmi; standarde; implementarea codoarelor de vorbire;*
- **Codarea audio ISO/MPEG:** *transmisia, stocarea, radiodifuzarea semnalelor audio; mascarea acustică; codarea sursă; codarea semnalelor audio; codecul audio ISO/MPEG; stereofonia multi-canal;*
- **Compresia semnalelor video și de imagine:** *tehnici bazate pe forma de undă; tehnici bazate pe modele; standarde; pre- și postprocesare;*
- **Compresia fără pierderi pentru text și imagini:** *modelare și codare; modele pentru text, modele pentru imagini.*

Editura **Teora**
TEH COMPRESIA SEMNALELOR
Cod: 0287

ISBN 973-20-0287-5
9 789732 002872